

**Titre:** Un algorithme primal de génération de colonnes pour le problème  
Title: du p-Centre-Somme

**Auteur:** Stéphane Alarie  
Author:

**Date:** 1997

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Alarie, S. (1997). Un algorithme primal de génération de colonnes pour le  
Citation: problème du p-Centre-Somme [Mémoire de maîtrise, École Polytechnique de  
Montréal]. PolyPublie. <https://publications.polymtl.ca/8968/>

 **Document en libre accès dans PolyPublie**  
Open Access document in PolyPublie

**URL de PolyPublie:**  
PolyPublie URL: <https://publications.polymtl.ca/8968/>

**Directeurs de  
recherche:**  
Advisors:

**Programme:** Non spécifié  
Program:



UNIVERSITÉ DE MONTRÉAL

UN ALGORITHME PRIMAL DE GÉNÉRATION DE COLONNES  
POUR LE PROBLÈME DU  $p$ -CENTRE-SOMME

STÉPHANE ALARIE  
DÉPARTEMENT DE MATHÉMATIQUES  
ET DE GÉNIE INDUSTRIEL  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(MATHÉMATIQUES APPLIQUÉES)

AVRIL 1997

© Stéphane Alarie, 1997.



National Library  
of Canada

Acquisitions and  
Bibliographic Services

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque nationale  
du Canada

Acquisitions et  
services bibliographiques

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file Votre référence*

*Our file Notre référence*

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-26448-3

Canada



UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

**UN ALGORITHME PRIMAL DE GÉNÉRATION DE COLONNES  
POUR LE PROBLÈME DU  $p$ -CENTRE-SOMME**

présenté par: ALARIE Stéphane

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. SOUMIS François, Ph.D., président

Mme JAUMARD Brigitte, T.Hab., membre et directrice de recherche

M. HANSEN Pierre, T.Agr., membre et codirecteur

M. LAPORTE Gilbert, Ph.D., membre

## RÉSUMÉ

Le problème du *p-Centre-Somme* est un problème de localisation qui se situe entre les deux problèmes du *p-Centre* et de la *p-Médiane*. Ce problème est défini comme suit: soit deux ensembles finis, l'un de  $n$  usagers et l'autre de  $m$  sites potentiels, localiser  $p$  facilités ( $p \leq m$ ) de sorte que l'affectation de chacun des  $n$  usagers à l'une des facilités minimise la somme des distances maximales entre les facilités et les usagers qui leur sont affectés. Nous comparons deux formulations mathématiques et proposons une méthode primale de résolution exacte par génération de colonnes. Des expériences de calcul montrent qu'on peut résoudre dans des temps raisonnables des instances de 50% de densité avec  $m = n = 1500$  et  $p \leq 150$ , c.-à-d. des problèmes avec un peu plus de 2 millions de variables et 1500 contraintes. Notre approche est flexible: elle est valide pour toute norme et la détermination du coût des affectations peut être donnée par n'importe quelle fonction. De plus, certaines contraintes de capacité et de passage peuvent être introduites sans détériorer la complexité de l'algorithme. On peut permettre aussi le choix de l'équipement des facilités par l'intermédiaire de fonctions coûts en escalier. Le problème du *p-Centre-Somme* se révèle un bon modèle pour la localisation des radars, des unités d'urgence et des antennes en téléphonie cellulaire.

## ABSTRACT

The *p*-Center-Sum problem is a location-allocation problem between the *p*-Center and the *p*-Median problems. This problem is defined as follows: given two finite sets, one of  $n$  users and another of  $m$  potential sites, locate  $p$  facilities (where  $p \leq m$ ) so that the allocation of each user to a facility minimizes the sum of the maximal distances between each facility to its users. We compare two mathematical formulations and propose an exact primal column generation algorithm. Computational experiences show that it is possible to solve within a reasonable computing time 50% density instances with up to  $m = n = 1500$  and  $p \leq 150$ , i.e. problems with up to 2 million variables and 1500 constraints. Our approach is flexible: it is valid for all norms and the determination of the allocation cost may be given by any function. Moreover, some kind of capacity and passage constraints can be added without worsening of the complexity of the algorithm. We can also choose the equipment of the facilities by the use of stairwise functions. The *p*-Center-Sum problem is an efficient model for the localisation of radars, emergency units and antennas in mobile cellular networks.



## TABLE DES MATIÈRES

RÉSUMÉ .....	iv
ABSTRACT .....	v
TABLE DES MATIÈRES.....	vi
LISTE DES TABLEAUX.....	xi
LISTE DES FIGURES.....	xii
INTRODUCTION.....	1
CHAPITRE 1 Description du problème.....	6
1.1 Définitions .....	6
1.2 Généralisation et propriétés.....	8
1.3 Formulations mathématiques.....	14



1.4 Dégénérescence des relaxations linéaires .....	19
1.5 Comparaison théorique des bornes .....	20
1.6 Propriétés spécifiques du $(PL_2)$ .....	26
 CHAPITRE 2 Résolution du problème .....	 34
2.1 Justification de la génération de colonnes .....	34
2.2 Décomposition .....	36
2.2.1 Problème maître .....	37
2.2.2 Problème auxiliaire .....	38
2.3 Propriétés et résolution du problème auxiliaire .....	40
2.3.1 Listes ordonnées et recouvrements d'usagers .....	40
2.3.2 Méthode de résolution .....	42
2.4 Résolution de la relaxation linéaire .....	46
2.5 Suggestion d'une énumération implicite .....	50



2.5.1 Règle de branchement .....	52
2.5.2 Sélection des usagers.....	53
2.5.3 Convergence de l'énumération.....	55
2.5.4 Qualité du branchement .....	57
2.5.5 Rapprochement avec les bornes supérieures généralisées .....	59
2.5.6 Stratégie de branchement .....	60
2.5.7 Modification de la matrice du (PM) .....	61
<b>2.6 Résolution du problème auxiliaire sous le branchement .....</b>	<b>61</b>
2.6.1 Problème auxiliaire sous un branchement général .....	62
2.6.2 Méthode de résolution .....	63
<b>CHAPITRE 3 Résultats .....</b>	<b>67</b>
3.1 Bancs d'expérimentation .....	67
3.2 Comparaison expérimentale des bornes .....	69
3.3 Comparaison avec la littérature.....	71



3.4 Performance de la règle de branchement.....	74
3.5 Effet du degré d'intersection entre $I$ et $J$ .....	78
3.6 Effet de la dimension des instances .....	81
 CHAPITRE 4 Complément et différentes extensions.....	 83
4.1 Extension aux coûts d'affectation non linéaires.....	83
4.1.1 Fonctions du type logarithmique et exponentiel.....	84
4.1.2 Fonctions du type par paliers .....	88
4.2 Intégration de contraintes à même les $L_j$ et les $E_j$ .....	91
4.3 Généralisation du modèle .....	93
4.3.1 Nouvel algorithme pour le problème de Recouvrement .....	94
4.3.2 Application au domaine de localisation continue .....	94
4.3.3 Interprétation étendue du modèle .....	95
 CONCLUSION .....	 97

<b>BIBLIOGRAPHIE .....</b>	<b>99</b>
----------------------------	-----------

## LISTE DES TABLEAUX

3.1	Comparaison du saut de dualité des formulations (PCS <sub>1</sub> ) et (PCS <sub>2</sub> ) ..	70
3.2	Comparaison avec la littérature .....	72
3.3	Comparaison de la performance des règles de branchement .....	75
3.4	Conditions de résolution selon le degré d'intersection $\rho$ entre $I$ et $J$ ..	79
3.5	Conditions de résolution selon la dimension des instances .....	81
4.1	Conditions de résolution pour la fonction logarithmique.....	85
4.2	Conditions de résolutions pour la fonction exponentielle .....	86
4.3	Conditions de résolution pour la fonction approchée par 3 paliers ....	90
4.4	Conditions de résolution pour la fonction approchée par 5 paliers ....	90

## LISTE DES FIGURES

1.1	Exemple illustrant la qualité des bornes de $(PL_1)$ et $(PL_2)$ .....	25
4.1	Fonctions logarithmique et exponentielle utilisées pour les expériences	84
4.2	Fonctions par paliers utilisées pour les expériences .....	89

# INTRODUCTION

Dans sa plus grande généralité, le problème de localisation consiste à déterminer l'emplacement d'un certain nombre de facilités ainsi qu'à décider l'affectation d'un ensemble d'utilisateurs à ces facilités de façon à optimiser une fonction sous un certain nombre de contraintes. La valeur de la fonction objectif dépend de la distance séparant les utilisateurs des sites potentiels.

Considérons les deux problèmes de localisation du *p-Centre* et de la *p-Médiane*. Le problème du *p-Centre* consiste à minimiser la valeur *maximale* de l'ensemble des distances *maximales* séparant chacune des  $p$  facilités aux utilisateurs qu'on leur a affectés. Ainsi, le problème du *p-Centre* a une fonction objective du type *p-Max-Max*. Pour ce qui est du problème de la *p-Médiane*, il s'agit de minimiser la *somme* de la *somme* des distances entre chacune des  $p$  facilités et les utilisateurs qui leur sont affectés. De ce fait, l'expression *p-Somme-Somme* peut aussi s'utiliser pour décrire la fonction objectif du problème de la *p-Médiane*.

Ces deux problèmes ont été étudiés sous différentes formes. Historiquement, les premières formes étaient du type *discret/continu*, cas où les utilisateurs forment un ensemble discret de points dans le plan et où tous les points de ce plan sont admissibles pour l'ouverture d'une facilité. Notons que cette forme inclut le problème classique de Weber comme cas particulier. Une seconde forme étudiée a été le cas *discret/discret* pour laquelle les sites potentiels des facilités sont aussi limités à un ensemble discret de points dans le plan. Ces deux formes ont été étudiées dans le cas particulier de graphes. Ainsi, un cas *discret/continu* a été défini en considérant les sommets comme des utilisateurs et tous les points du graphe, aussi bien

ceux appartenant aux arêtes que ceux constituant les sommets, à des sites potentiels. Pour le cas *discret/discret*, on restreint en plus les sites potentiels aux seuls sommets du graphe. Une forme *continu/discret* a aussi été définie sur les graphes, il s'agit alors de considérer tous les points appartenant au graphe comme des usagers que l'on doit desservir à partir de facilités situées seulement sur des sommets du graphe. Le cas général *continu/discret* n'a encore fait l'objet d'aucune étude. De plus, il reste une dernière variation inexplorée: celle du type *continu/continu* où usagers et sites potentiels existent sur des domaines continus.

Hansen, Labbé et Minoux (1994) ont observé que par la permutation des opérateurs *Max* et *Somme*, on pouvait obtenir deux nouvelles fonctions objectifs. Ainsi, à l'objectif *p-Max-Somme* correspond le problème de la *p-Médiane-Max* qui consiste à minimiser la valeur *maximale* de la *somme* des distances entre chacune des  $p$  facilités aux usagers qui leur sont affectés. Ce dernier problème peut servir à déterminer un ensemble de tournées de véhicules dont la charge de travail des différentes tournées soit la plus homogène possible. L'autre fonction objectif, *p-Somme-Max*, définit le problème du *p-Centre-Somme*.

Le présent mémoire est consacré au problème du *p-Centre-Somme* sous sa forme *discrète/discrète*. Ce problème a été présenté dans Hansen, Labbé et Minoux (1994) comme suit: soit deux ensembles finis, l'un de  $n$  usagers et l'autre de  $m$  sites potentiels, le problème du *p-Centre-Somme* consiste à localiser  $p$  facilités ( $p \leq m$ ) de sorte que l'affectation de chacun des  $n$  usagers à l'une des facilités minimise la *somme* des distances *maximales* entre les facilités et les usagers qui leur sont affectés.

Le problème du *p-Centre-Somme* se révèle un modèle intéressant pour minimiser la

couverture des radars en radiotélécommunication, en radiodétection ainsi qu'en radiobalisateur. On retrouve une application semblable dans le positionnement des satellites pour la diffusion à grande échelle d'émissions télévisuelles. En santé, le modèle peut s'appliquer à la localisation d'unités d'urgence avec une flotte hétérogène de véhicules afin de desservir un territoire dont la densité de la population n'est pas homogène. En supposant une discrétisation des usagers à desservir (le domaine initial des usagers étant continu *a priori*), une autre application intervient en téléphonie cellulaire. On doit alors décider l'emplacement des antennes parmi un ensemble prédéterminé de sites potentiels en plus de choisir le type d'antennes à installer sur les sites retenus. Chacun des types d'antennes est défini par un rayon de couverture qui lui est propre et son coût est fonction du rayon de l'antenne.

Hansen, Minoux et Labbé (1987) ont brièvement discuté de ce problème lors d'une note sur une extension de la génération de colonnes aux programmes mixtes en exploitant les propriétés de l'algorithme dual du simplexe. Ainsi, la méthode de résolution pour le problème du *p-Centre-Somme* que ces auteurs nous proposaient en 1994 dérive directement de leur note de 1987.

Comme la méthode primale de génération de colonnes était déjà appliquée en 1991 pour résoudre certains problèmes en nombres entiers (soit par exemple Desrosiers, Soumis et Desrochers, 1984), Hansen, Jaumard et Poggi de Aragão ont alors repris l'extension de la méthode de génération de colonnes aux programmes mixtes avec l'algorithme primal du simplexe. Ces trois derniers auteurs ont comparé leur méthode avec celle développée par Hansen, Minoux et Labbé en 1987 sur le problème de *Satisfaisabilité Probabiliste Maximale* (Hansen, Jaumard et Poggi de Aragão, 1992), un problème combinatoire ayant des applications en intelligence artificielle. Expérimentalement, Hansen, Jaumard et Poggi de Aragão ont démontré alors qu'une



méthode primale de résolution par génération de colonnes est supérieure à une méthode duale.

La méthode primale de résolution par génération de colonnes a déjà été appliquée pour résoudre la forme *discrète/discrète* du problème de la *p-Médiane* sous des contraintes de capacité (Hansen, Jaumard et Sanlaville, 1994). Nous proposons donc de reprendre la résolution de la forme *discrète/discrète* du problème du *p-Centre-Somme* par une méthode primale de génération de colonnes afin d'améliorer substantiellement les résultats obtenus par Hansen, Labbé et Minoux en 1994. Ainsi, nous montrerons qu'avec cette approche nous pouvons résoudre facilement des instances ayant jusqu'à 1500 usagers et autant de sites potentiels alors que la méthode duale ne permettait pas d'aller au-delà de 50 usagers et 50 sites.

Étant donné qu'un problème de localisation peut souvent se réinterpréter comme un problème de classification, la méthode que nous présentons est aussi utilisable pour résoudre le problème équivalent du *p-Centre-Somme* en classification, soit le problème de la *Somme Minimale des Rayons*. Afin de passer d'un contexte à l'autre, il suffit de considérer que les distances correspondent aux dissimilarités, les usagers aux objets, les affectations aux classes et l'ouverture d'une facilité sur un site potentiel au choix du représentant d'une classe. Remarquons que les problèmes de classification sont plus généraux que ceux de la localisation puisque les dissimilarités entre deux objets ne sont pas toujours symétriques et ne correspondent pas nécessairement à des distances. Dans la suite de ce mémoire, nous ne poserons donc aucune hypothèse particulière sur la fonction mesurant la distance entre un usager et un site potentiel.

Le premier chapitre du mémoire présente la définition du problème du *p-Centre-Somme* telle que formulée par Hansen, Labbé et Minoux (1994) sous sa forme

*discrète/discrète* ainsi que les deux formulations mathématiques proposées alors par les auteurs. Une comparaison des bornes inférieures provenant de la relaxation linéaire des formulations est ensuite donnée en plus de quelques propriétés intrinsèques au problème du *p-Centre-Somme*. Un algorithme primal de génération de colonnes est présenté par la suite au chapitre 2, lequel utilise dans un contexte de recouvrement une règle de branchement du même type que celle de Ryan et Foster (1981) alors que dans la littérature, ce type de branchement est réservé à la partition (Barnhart *et al.*, 1994; Vanderbeck et Wolsey, 1996). Le chapitre 3 suit avec quelques résultats numériques. Les performances de notre méthode de résolution sont comparées avec celles de la méthode proposée par Hansen, Labbé et Minoux en 1994. Le chapitre 4 étend le coût des affectations aux fonctions non linéaires et considère l'ajout de nouvelles contraintes au problème du *p-Centre-Somme*. Enfin, une brève conclusion termine le mémoire.

# CHAPITRE 1

## Description du problème

### 1.1 Définitions

Soit  $I = \{1, \dots, i, \dots, n\}$  un ensemble de  $n$  usagers et  $J = \{1, \dots, j, \dots, m\}$  un ensemble de  $m$  sites potentiels où une facilité peut être ouverte. Prenons  $p \leq m$  comme étant le nombre maximal de facilités que l'on est prêt à ouvrir pour desservir les  $n$  usagers. Ainsi, nous acceptons de desservir les usagers avec moins de  $p$  facilités si cela est possible à un moindre coût. Notons qu'un usager est entièrement ou pas du tout desservi par une facilité, et que  $d_{ij}$  représente la distance d'un usager  $i$  à un site potentiel  $j$ .

Dans notre souci d'obtenir la plus grande généralité possible, nous ne limiterons pas l'existence des éléments  $i$  et  $j$  aux seuls espaces métriques. Ainsi, on ne fait pas d'hypothèses sur l'utilisation d'une norme en particulier, ni d'une symétrie des distances entre un usager et un site, ni entre deux usagers ou deux sites. De plus, on n'exige pas que l'inégalité triangulaire  $d_{ik} \leq d_{ij} + d_{jk}$  soit satisfaite pour tout  $i, j, k \in I \cup J$ . En fait, on peut considérer que les valeurs  $d_{ij}$  pour un usager  $i$  sont données par une fonction d'insatisfaction  $\bar{s}_i(j)$  pour un site  $j$ . Comme  $\bar{s}_i(j) = d_{ij}$ , la seule exigence que nous imposons à  $\bar{s}_i(j)$  est de satisfaire  $d_{ij} \geq 0$  où  $d_{ij} = 0$  pour tout usager  $i$  situé sur le site  $j$ .

Le problème du *p-Centre-Somme* a été défini par Hansen, Labbé et Minoux (1994) de la façon suivante: trouver un sous-ensemble  $J_p^* = \{j_1^*, \dots, j_p^*\}$  de  $J$  et une partition

de  $I$  en  $p$  sous-ensembles  $I_1^*, \dots, I_p^*$  tels que

$$\sum_{k=1}^p \left( \max_{i \in I_k^*} d_{ij_k^*} \right) \leq \sum_{k=1}^p \left( \max_{i \in I_k} d_{ij_k} \right) \quad (1.1)$$

pour tout sous-ensemble  $\{j_1, \dots, j_p\}$  de  $J$  et pour toute partition  $\{I_1, \dots, I_p\}$  de  $I$ . Le sous-ensemble optimal  $J_p^*$  porte le nom de *p-Centre-Somme*. Noter que le cas particulier où l'on utilise  $q$  facilités ( $q < p$ ) est couvert par cette définition en prenant  $I_{q+1}^* = I_{q+2}^* = \dots = I_p^* = \emptyset$  et en convenant que  $\max_{i \in I_k^*} d_{ij_k^*} = 0$  quand  $I_k^* = \emptyset$ .

Nous donnons donc priorité dans cette définition du problème à la satisfaction naturelle des usagers d'être affectés à une facilité ouverte sur un site donné plutôt que sur un autre. Ainsi, nous considérons que plus un usager  $i$  est éloigné d'un site  $j$ , plus il sera insatisfait d'être affecté à la facilité qui y sera ouverte et plus il sera coûteux d'améliorer sa satisfaction.

Considérons maintenant le cas particulier où une seule facilité est ouverte ( $p = 1$ ), alors  $J_1^* = \{j^*\}$  et par conséquent  $I$  est partitionné en un seul sous-ensemble  $I^*$ , ce qui implique  $I = I^*$ . L'équation (1.1) se réduit alors à

$$\max_{i \in I} d_{ij^*} \leq \max_{i \in I} d_{ij}$$

ce qui correspond au problème du *1-Centre* où  $j^*$  est le *1-Centre* de  $I$ . Le problème du *p-Centre-Somme* est ainsi une généralisation du problème du *1-Centre*.

On peut reformuler le problème du *1-Centre* sous la forme équivalente du problème de classification du *Cercle de Rayon Minimum* où il s'agit de recouvrir tous les éléments  $i$  de  $I$  avec un cercle de rayon  $r_{j^*}$  tel que  $j^* \in J$  est le centre du cercle et  $r_{j^*} \leq r_j$  pour tout  $j \in J$ , ce qui se réécrit

$$r_{j^*} = \min_{j^*} \max_{i \in I} d_{ij^*}$$

où les  $d_{ij}$  correspondent maintenant aux coefficients de dissimilarité.

Un cercle de rayon  $r_{j^*}$  est dit critique puisqu'au moins un usager  $i$  se situe nécessairement sur sa circonférence. Pour cette raison, nous dirons de tout usager appartenant à la circonférence d'un cercle critique qu'il est critique et nous le noterons  $\hat{i}_{j^*}$ . Formellement,  $i = \hat{i}_{j^*}$  si et seulement si  $r_{j^*} = d_{ij^*}$  pour un  $j^*$  donné. En conséquence, les usagers critiques  $\hat{i}_{j^*}$  correspondent aux usagers les plus éloignés du sous-ensemble  $I^*$  centrée au site  $j^*$ .

## 1.2 Généralisation et propriétés

On considère la généralisation suivante, notée (PCS), du problème du *p-Centre-Somme*: trouver un sous-ensemble  $J_p^* = \{j_1^*, \dots, j_p^*\}$  de  $J$  et un recouvrement de  $I$  en  $p$  sous-ensembles  $I_1^*, \dots, I_p^*$  tels que

$$(PCS) \quad \sum_{k=1}^p \left( \max_{i \in I_k^*} d_{ij_k^*} \right) \leq \sum_{k=1}^p \left( \max_{i \in I_k} d_{ij_k} \right) \quad (1.2)$$

pour tout sous-ensemble  $\{j_1, \dots, j_p\}$  de  $J$  et pour tout recouvrement  $\{I_1, \dots, I_p\}$  de  $I$ . Comme précédemment, on admet que certains sous-ensembles  $I_k^*$  du recouvrement optimal puissent être vides et qu'alors  $\max_{i \in I_k^*} d_{ij_k^*} = 0$ .

Nous montrerons à la fin de cette section (théorème 1.6) que les deux problèmes (1.1) et (1.2) ont la même valeur optimale et que l'on peut aisément passer d'une solution optimale de l'un à une solution optimale de l'autre. Pour cette raison, sauf indication spécifique, nous désignerons par problème du *p-Centre-Somme* la version généralisée définie par (1.2).

Les résultats de complexité obtenus par Hansen, Labbé et Minoux (1994) pour la version partitionnement du problème du *p-Centre-Somme* s'étendent immédiatement à la version recouvrement:

**Proposition 1.1** *Le problème du p-Centre-Somme est NP-dur.*

**Proposition 1.2** *Lorsque le p-Centre-Somme  $J_p^*$  est donné, l'affectation des usagers aux sites  $j_k^*$  qui minimise la somme des  $r_{j_k^*}$  est NP-complet.*

**Preuve:** La version partitionnement (1.1) du problème du *p-Centre-Somme* est un cas particulier de sa version recouvrement (1.2). □

La proposition 1.2 illustre une différence fondamentale entre le problème du *p-Centre-Somme* et ceux du *p-Centre* et de la *p-Médiane*. Pour ces deux derniers problèmes, on peut obtenir la solution optimale en temps polynomial lorsque les facilités sont fixées puisqu'il suffit alors d'affecter les usagers à la facilité la plus proche. La proposition 1.2 indique qu'il en est autrement pour le problème du *p-Centre-Somme*. D'ailleurs, Hansen, Labbé et Minoux (1994) fournissent un exemple illustrant que l'affectation des usagers à la facilité la plus proche peut ne pas être optimale.

Nous présentons maintenant différentes propriétés des solutions optimales du (PCS). Notre premier résultat permet l'interprétation du problème en termes d'un recouvrement de  $I$  par un ensemble de cercles critiques centrés sur les éléments de  $J_p^*$ .

**Proposition 1.3** *Chaque site  $j^* \in J_p^*$  est un 1-Centre optimal du sous-ensemble  $I^*$  correspondant.*

**Preuve:** Supposons qu'il existe un site  $j \in J$  distinct de  $j^* \in J_p^*$  tel que

$$r_{j^*} = \max_{i \in I^*} d_{ij^*} > r_j = \max_{i \in I^*} d_{ij}.$$

Puisque la solution réalisable obtenue en changeant  $j^*$  par  $j$  a une valeur inférieure, il s'ensuit alors une contradiction même si  $j \in J_p^*$ .  $\square$

De cette proposition, on déduit que lorsque les sous-ensembles  $I^*$  d'un recouvrement optimal du (PCS) sont connus, les éléments de  $J_p^*$  s'obtiennent en résolvant un problème du 1-Centre sur chaque  $I^*$ . Ceci revient donc à chercher le plus petit cercle critique recouvrant complètement le sous-ensemble  $I^*$ . Cependant, les sous-ensembles  $I^*$  ne sont habituellement pas connus d'avance. Par contre, nous allons montrer que les sous-ensembles potentiels  $I^*$  peuvent être déduits en recouvrant  $I$  par des cercles critiques dont on veut minimiser la somme des rayons.

Recouvrons un sous-ensemble  $I^*$  avec un cercle quelconque dont aucun des éléments ne se trouve sur la circonférence. On peut abaisser le coût du recouvrement en diminuant le rayon jusqu'à ce que la circonférence du cercle passe par un élément de  $I^*$ . Le cercle est alors critique. Naturellement, on peut construire ainsi une infinité de cercles critiques recouvrant  $I^*$ . Si  $J$  est continu, Chen et Handler (1987) ont montré que le nombre de cercles critiques à considérer pour recouvrir  $I$  en minimisant le coût est  $\binom{n}{3} + \binom{n}{2} + n$ , où  $\binom{n}{3}$  est le nombre de cercles définis avec trois usagers sur la circonférence,  $\binom{n}{2}$  est le nombre de cercles dont le diamètre est déterminé par deux usagers, et  $n$  est le nombre de cercles de rayon nul (cas d'une facilité desservant un seul usager). Pour  $J$  discret, le nombre de cercles critiques est

encore plus réduit (on suppose  $m \leq n$ ). Le résultat suivant nous fournit d'ailleurs une borne supérieure sur le nombre de cercles critiques à considérer.

**Proposition 1.4** *Pour un domaine discret  $J$  de localisation des facilités, il y a au plus  $mn$  cercles critiques possibles.*

**Preuve:** Prenons  $j \in J$  et considérons le nombre de cercles critiques différents centrés en  $j$  que l'on peut obtenir. Si tous les usagers  $i$  sont à des distances  $d_{ij}$  différentes, alors nous obtenons  $n$  cercles possibles. Par contre, pour deux usagers à la même distance de  $j$ , nous n'avons qu'un seul cercle critique. Il y a donc au plus  $n$  cercles critiques par site de  $J$ . Comme  $|J| = m$  où  $|J|$  désigne la cardinalité de  $J$ , il y a alors au plus  $mn$  cercles possibles.  $\square$

Remarquons que les usagers critiques  $i_{j^*}$  appartenant à un sous-ensemble  $I^*$  centré en  $j^*$  sont les seuls usagers à intervenir dans le calcul du coût de  $I^*$ . De plus, il suffit de considérer un seul  $i_{j^*}$  pour que le coût de  $I^*$  soit déterminé. Ceci nous amène à envisager la possibilité de construire à partir d'un recouvrement optimal  $\{I_1^*, \dots, I_p^*\}$  de  $I$  des solutions de même valeur telles qu'un usager puisse être retiré d'un sous-ensemble tout en maintenant une solution réalisable. Le prochain résultat implique qu'il existe dans tout recouvrement optimal de  $I$  un ensemble d'usagers critiques tel que aucun de ceux-ci ne pourrait être retiré d'un sous-ensemble de la solution sans la rendre irréalisable.

**Proposition 1.5** *Dans toute solution optimale du problème du  $p$ -Centre-Somme, chaque sous-ensemble  $I^*$  possède au moins un usager critique qui n'appartient à aucun autre sous-ensemble  $I^*$  de la solution.*



**Preuve:** Soit un recouvrement optimal  $\{I_1^*, I_2^*, \dots, I_p^*\}$  du (PCS) ayant pour valeur  $z^*$ . Supposons, sans perte de généralité, que chaque usager critique  $i_{j_1^*}$  de  $I_1^*$  appartient aussi à au moins un autre sous-ensemble  $I_k^*$  où  $k = 2, \dots, p$ . On peut alors construire un recouvrement réalisable  $\{I_1^\dagger, I_2^*, \dots, I_p^*\}$  de valeur  $z^\dagger$  où  $I_1^\dagger$  est le sous-ensemble  $I_1^*$  privé de ses  $i_{j_1^*}$ . Comme  $z^\dagger < z^*$  puisque le rayon de  $I_1^\dagger$  est strictement plus petit que celui de  $I_1^*$ , l'optimalité de  $z^*$  est ainsi contredite. Le cas où  $I_1^\dagger$  serait vide correspondrait à la fermeture de la facilité ouverte précédemment en  $j_1^*$ . La proposition est donc démontrée.  $\square$

On peut maintenant formuler le résultat qui permet la généralisation annoncée au début de la présente section.

**Théorème 1.6** *À toute solution optimale du problème du  $p$ -Centre-Somme obtenue à partir d'une partition de  $I$  correspond une solution optimale du problème du  $p$ -Centre-Somme obtenue à partir d'un recouvrement de  $I$  de même valeur et réciproquement.*

**Preuve:**

$\Rightarrow$  À partir d'une solution optimale du (PCS) de valeur  $z^*$  d'un recouvrement de  $I$ , on peut construire une solution réalisable pour une partition de  $I$  de même valeur par une élimination successive des superpositions entre les sous-ensembles d'utilisateurs formant le recouvrement. Ainsi, on prend le premier sous-ensemble et tout utilisateur de ce sous-ensemble qui est présent dans un autre sous-ensemble est retiré de ce dernier. On reprend avec un deuxième sous-ensemble, puis un troisième et ainsi de suite jusqu'à ce qu'on ait considéré

tous les sous-ensembles. Comme la proposition 1.5 nous assure qu'on ne pourra jamais retirer tous les usagers critiques d'un sous-ensemble puisqu'il existe dans chaque sous-ensemble au moins un usager critique qui lui soit exclusif, la valeur de la fonction objective reste donc inchangée.

⇐ D'autre part, la solution optimale du (PCS) de valeur  $z^*$  d'une partition de  $I$  est nécessairement une solution réalisable comme recouvrement de  $I$  puisque la partition est un cas particulier du recouvrement. L'implication dans les deux sens est ainsi démontrée.  $\square$

Le résultat qui suit découle directement du théorème 1.6 et il est donc cité sans preuve.

**Corollaire 1.7** *Le  $p$ -Centre-Somme  $J_p^*$  reste le même aussi bien pour une partition que pour une couverture de  $I$ .*

Notons que  $J_p^*$  n'est pas nécessairement unique. Ceci n'est aucunement gênant puisqu'ils sont tous équivalents étant donné qu'ils satisfont l'inégalité (1.2).

De plus, remarquons qu'intuitivement le corollaire 1.7 est aussi valide pour une version de mise en boîte (*packing*) du problème du  $p$ -Centre-Somme (version qui consisterait à maximiser la sommation des  $r_{j^*}$  où  $j^* \in J_p^*$  sans recouvrir plus d'une fois chaque usager de  $I$ ). Pour s'en convaincre, partons d'un recouvrement optimal de  $I$  par des cercles critiques et diminuons les uns après les autres les intersections entre les cercles jusqu'à ce qu'elles soient vides en prenant à chaque itération le cercle qui minimise la diminution de la valeur objective lorsque l'on réduit le rayon de ce dernier jusqu'au cercle critique suivant.

### 1.3 Formulations mathématiques

La première formulation mathématique, notée  $(PCS_1)$ , que nous présentons exprime  $(PCS)$  comme un programme linéaire mixte en variables 0-1 avec  $mn + 2m$  variables et  $2mn + n + 1$  contraintes. Remarquons qu'aucun algorithme exact n'a encore été proposé pour résoudre  $(PCS_1)$ .

$$\min \quad \sum_{j=1}^m z_j \quad (1.3)$$

$$\text{s.c.} \quad z_j \geq d_{ij}x_{ij} \quad i = 1, 2, \dots, n; j = 1, 2, \dots, m \quad (1.4)$$

$$\sum_{j=1}^m x_{ij} \geq 1 \quad i = 1, 2, \dots, n \quad (1.5)$$

$$(PCS_1) \quad x_{ij} \leq y_j \quad i = 1, 2, \dots, n; j = 1, 2, \dots, m \quad (1.6)$$

$$\sum_{j=1}^m y_j \leq p \quad (1.7)$$

$$x_{ij} \in \{0, 1\} \quad i = 1, 2, \dots, n; j = 1, 2, \dots, m \quad (1.8)$$

$$y_j \in \{0, 1\} \quad j = 1, 2, \dots, m \quad (1.9)$$

La variable  $x_{ij}$  est égale à 1 si l'utilisateur  $i$  est affecté au site  $j$  et à 0 sinon. La variable  $y_j$  est égale à 1 si une facilité est ouverte sur le site  $j$  et à 0 sinon. La variable  $z_j$  est égale à la distance maximale entre le site  $j$  et les usagers qui lui ont été affectés. Si aucun usager n'est affecté en  $j$  alors  $z_j = 0$ . De ce fait,  $z_j \geq 0$  est implicite pour tout  $j \in J$ . De plus, on dit de (1.4) qu'elles sont des contraintes de seuil. Les contraintes (1.5) impliquent que chaque usager  $i$  soit affecté à au moins un site  $j$ . Les contraintes (1.6) imposent qu'un usager  $i$  puisse être affecté à un site si et

seulement si une facilité a été ouverte sur ce site. La contrainte (1.7) restreint le nombre maximal de facilités pouvant être ouvertes à  $p$ . Finalement, les contraintes (1.8) et (1.9) obligent que les facilités soient entièrement ouvertes ou pas du tout et que les usagers soient entièrement affectés ou pas du tout aux facilités. La validité de la formulation  $(PCS_1)$  est donnée par la proposition qui suit.

**Proposition 1.8** *Résoudre  $(PCS_1)$  est équivalent à résoudre  $(PCS)$ .*

**Preuve:**

$\Rightarrow$  Soit une solution optimale du  $(PCS)$  de valeur  $z^*$ , pour  $j \in J_p^*$ , poser  $y_j = 1$  et pour  $i \in I^*$  où  $I^*$  désigne l'ensemble des usagers affectés à  $j$ , poser  $x_{ij} = 1$ . Dans le cas contraire, poser  $x_{ij}$  et  $y_j = 0$ . Les contraintes (1.5), (1.6) et (1.7) sont alors satisfaites. Les  $z_j$  s'obtiennent trivialement puisque  $x_{ij}$  et  $y_j$  sont fixés. Nous obtenons ainsi une solution réalisable du  $(PCS_1)$  de même valeur  $z^*$  puisque pour chaque  $j \in J_p^*$  ainsi que pour tout  $I^*$  associé à chacun de ces sites  $j$ , nous avons

$$z_j = \max_{i \in I} \{d_{ij}x_{ij}\} = \max_{i \in I^*} d_{ij}.$$

$\Leftarrow$  Considérons une solution optimale du  $(PCS_1)$  de valeur  $z_1^*$ . On veut montrer que  $z_1^* \geq z^*$  où  $z^*$  est la valeur de la solution réalisable du  $(PCS_1)$  correspondant à la solution optimale du  $(PCS)$ . Supposons que  $z_1^* < z^*$ . Comme  $z^*$  est la valeur optimale du  $(PCS)$ , il est donc impossible de trouver une solution réalisable du  $(PCS)$  de valeur  $z_1^*$ , d'où une contradiction. Puisque l'on ne peut obtenir une valeur du  $(PCS_1)$  inférieure à  $z^*$ , il s'ensuit que  $z^*$  est aussi la valeur optimale du  $(PCS_1)$ .  $\square$

Considérons maintenant la représentation d'un sous-ensemble optimal  $I^*$  sous la

forme d'un vecteur-colonne de dimension  $n$  où chacune des lignes du vecteur est associée à un et un seul usager de  $I$ . Pour tout  $i \in I$ , nous avons un 1 sur la ligne correspondante à  $i$  si  $i \in I^*$  et un 0 autrement. Ainsi, nous pouvons définir  $K_j$  comme étant l'ensemble des vecteurs-colonnes non vides représentant tous les recouvrements possibles d'utilisateurs au site  $j$ . Puisqu'à tout recouvrement d'utilisateurs correspond un et un seul cercle critique, il y a donc autant de colonnes dans  $K_j$  que de cercles critiques centrés en  $j$ . Plus précisément, il y a pour chaque cercle critique centré sur  $j$  qu'une et une seule colonne dans  $K_j$ . La démonstration de la proposition 1.4 donne alors  $|K_j| \leq n$  où  $|K_j|$  désigne la cardinalité de  $K_j$ .

Soit l'ensemble  $K = \{1, \dots, k, \dots, \ell\}$  avec  $\ell \leq mn$ , l'ensemble des colonnes non vides des recouvrements possibles aux différents sites de  $J$  résultant de l'union des  $K_j$ . On obtient alors une deuxième formulation, notée  $(PCS_2)$ , qui est un programme en nombres entiers avec  $\ell$  variables et  $n + 1$  contraintes. Il s'agit en fait d'un problème de recouvrement avec une limitation sur le nombre de cercles critiques.

$$\min \quad \sum_{k=1}^{\ell} c_k y_k \quad (1.10)$$

$$(PCS_2) \quad \text{s.c.} \quad \sum_{k=1}^{\ell} a_{ik} y_k \geq 1 \quad i = 1, 2, \dots, n \quad (1.11)$$

$$\sum_{k=1}^{\ell} y_k \leq p \quad (1.12)$$

$$y_k \in \{0, 1\} \quad k = 1, 2, \dots, \ell \quad (1.13)$$

La variable  $y_k$  est égale à 1 si le recouvrement de la colonne  $k$  est retenue comme

affectation d'utilisateurs et à 0 sinon. Un recouvrement d'utilisateurs pour une colonne  $k$  est définie par  $a_{ik} = 1$  si l'utilisateur  $i$  est affecté au site associé avec  $k$  et  $a_{ik} = 0$  autrement. Dans l'objectif (1.10), le coût  $c_k$  d'une colonne correspond à la distance maximale entre le site  $j$  associé à  $k$  et les utilisateurs qui lui ont été affectés, soit  $c_k = \max_{i \in I} \{a_{ik} d_{ij}\}$ . Comme  $c_k$  correspond à la valeur du rayon du cercle critique représenté par la colonne  $k$ , nous dirons des utilisateurs pouvant déterminer  $c_k$  (soit les utilisateurs  $i \in \operatorname{argmax}_{i \in I} \{a_{ik} d_{ij}\}$ ) qu'ils sont les utilisateurs critiques de  $k$ . Les contraintes (1.11) imposent que chaque utilisateur soit recouvert par au moins une colonne et la contrainte (1.12) limite à  $p$  le nombre de colonnes qui peuvent être retenues. Enfin, par les contraintes (1.13), on s'assure que le choix d'un site et de l'affectation des utilisateurs à ce site soit réalisé complètement ou pas du tout. Notons qu'il est sous-entendu qu'une facilité soit ouverte sur le site associé à une colonne que l'on aura retenue. Comme précédemment,  $(PCS_2)$  est aussi une formulation valide du  $(PCS)$ .

**Proposition 1.9** *Résoudre  $(PCS_2)$  est équivalent à résoudre  $(PCS)$ .*

**Preuve:**

$\Rightarrow$  Soit une solution optimale du  $(PCS)$  de valeur  $z^*$ . Pour chaque  $j \in J_p^*$ , considérer les colonnes  $k \in K_j$  et poser  $y_k = 1$  si  $a_{ik} = 1$  pour tout utilisateur  $i \in I^*$  ( $I^*$  étant l'affectation optimale des utilisateurs au site  $j$ ) et si  $a_{ik} = 0$  pour tout  $i \notin I^*$ . Autrement, poser  $y_k = 0$ . Ainsi, nous obtenons une solution réalisable du  $(PCS_2)$  puisque toutes les contraintes (1.11) et (1.12) sont satisfaites. De plus, cette solution a aussi pour valeur  $z^*$  puisque pour la colonne  $k \in K_j$  telle que  $y_k = 1$ , nous avons

$$c_k = \max_{i \in I} \{a_{ik} d_{ij}\} = \max_{i \in I^*} d_{ij}.$$

$\Leftarrow$  Considérons une solution optimale du  $(PCS_2)$  de valeur  $z_2^*$ . On veut montrer que  $z_2^* \geq z^*$  où  $z^*$  est la valeur de la solution réalisable du  $(PCS_2)$  correspondant à la solution optimale du  $(PCS)$ . Supposons que  $z_2^* < z^*$ . Comme  $z^*$  est la valeur optimale du  $(PCS)$ , il est donc impossible de trouver une solution réalisable du  $(PCS)$  de valeur  $z_2^*$ , d'où une contradiction. Puisque l'on ne peut obtenir une valeur du  $(PCS_2)$  inférieure à  $z^*$ , il s'ensuit que  $z^*$  est aussi la valeur optimale du  $(PCS_2)$ .  $\square$

Observons que la dernière formulation admet qu'une solution soit constituée de plusieurs colonnes prises dans un même sous-ensemble  $K_j$ . En d'autres mots, une solution telle que  $y_{k_1} = y_{k_2} = 1$  avec  $k_1 \neq k_2$  et où  $k_1$  et  $k_2 \in K_j$  est réalisable pour  $(PCS_2)$ . Or cette situation ne peut pas arriver pour une solution optimale.

**Proposition 1.10** *À l'optimum du  $(PCS_2)$ , l'affectation des usagers à une facilité située en  $j$  ne peut être représentée que par au plus une seule colonne.*

**Preuve:** Soit une solution optimale  $y^*$  du  $(PCS_2)$  ayant  $y_{k_1}^* = y_{k_2}^* = 1$ , où les deux colonnes d'indices  $k_1$  et  $k_2$  sont associées à un même site. Notons que la colonne d'indice  $k_3$  associée à ce site ayant pour coût  $\max\{c_{k_1}, c_{k_2}\}$  et pour affectation tous les usagers des deux colonnes d'indices  $k_1$  et  $k_2$ , soit  $a_{ik_3} = \max\{a_{ik_1}, a_{ik_2}\}$  pour tout  $i$ , n'a pas été retenue dans la solution  $y^*$  ( $y_{k_3}^* = 0$ ). Considérons la solution obtenue avec le vecteur  $\hat{y}$  tel que  $\hat{y}_{k_1} = \hat{y}_{k_2} = 0$ ,  $\hat{y}_{k_3} = 1$  et  $\hat{y}_k = y_k^*$  pour tous  $k \neq k_1, k_2, k_3$ . Cette solution est réalisable et de coût inférieur à la solution fournie par  $y^*$ . Ceci contredit l'optimalité de  $y^*$  et donc, la proposition est vraie.  $\square$



## 1.4 Dégénérescence des relaxations linéaires

Nommons  $(PL_1)$  et  $(PL_2)$  les relaxations linéaires respectives du  $(PCS_1)$  et du  $(PCS_2)$ . Plus précisément, nous allons considérer la relaxation qui consiste à remplacer l'ensemble binaire  $\{0, 1\}$  des contraintes (1.8), (1.9) et (1.13) par l'intervalle fermé  $[0, 1]$ . Puisque  $(PCS_1)$  et  $(PCS_2)$  sont équivalents à  $(PCS)$  (propositions 1.8 et 1.9), il en résulte que  $(PL_1)$  et  $(PL_2)$  sont aussi des relaxations du  $(PCS)$ .

Puisque (1.4) représente des contraintes de seuil, il s'ensuit que ces contraintes sont sujettes à introduire une forte dégénérescence lors de la résolution du  $(PL_1)$  par un algorithme du simplexe. Ceci s'explique par le fait que, dans la mesure où  $z_j$  ne change pas de valeur, on peut être amené à considérer inutilement un nombre important de solutions. Or notons que beaucoup de ces solutions inutiles ne sont pas considérées dans la résolution du  $(PL_2)$ . Par définition, toute variable  $y_k$  du  $(PL_2)$  ne peut être associée qu'à un et un seul des recouvrements engendrés par les cercles critiques admissibles de la solution en nombres entiers. Toutes les autres affectations d'utilisateurs étant interdites dans la formulation  $(PL_2)$ , il faut donc s'attendre à obtenir une dégénérescence plus importante avec  $(PL_1)$  que  $(PL_2)$  puisqu'une telle restriction n'existe pas dans  $(PL_1)$ . En pratique, nous avons observé que la résolution de  $(PL_1)$  était beaucoup plus difficile que  $(PL_2)$  à ce niveau. Nous reviendrons sur ce point au chapitre 3.

D'autre part, on peut se demander pourquoi favoriser  $(PCS_2)$  sous sa version recouvrement au lieu de celle de partitionnement puisque d'après le théorème 1.6 les valeurs optimales coïncident. En fait, la réponse est donnée par la proposition 1.4. Une version partitionnement du  $(PCS_2)$  impliquerait  $m(2^n - 1)$  colonnes au lieu des  $mn$  colonnes du recouvrement puisqu'alors toutes les combinaisons non vides



d'utilisateurs sont admissibles comme affectation à un site. Or la proposition 1.4 implique qu'il n'y aurait pas plus de  $mn$  coûts possibles pour ces  $m(2^n - 1)$  colonnes. (Notons que cette borne supérieure est bien moins précise que ce que la proposition 1.4 peut laisser supposer puisque des cercles de centres différents mais de rayons identiques ont nécessairement le même coût.) Ainsi, la version partitionnement du  $(PL_2)$  est aussi une source importante de dégénérescence.

En conséquence, parmi toutes les formulations mathématiques considérées,  $(PCS_2)$  est celle qui minimise la dégénérescence de l'algorithme du simplexe lorsque l'on résout sa relaxation linéaire. Nous allons maintenant montrer que l'intérêt de la formulation  $(PCS_2)$  ne se réduit pas seulement à la faible dégénérescence du  $(PL_2)$  mais aussi à la qualité de la borne donnée par cette dernière.

## 1.5 Comparaison théorique des bornes

Selon Hansen, Labbé et Minoux (1994), la borne inférieure donnée par  $(PL_1)$  est pauvre. Leur affirmation repose sur le fait que pour des  $x_{ij}$  et  $y_j$  de valeurs fractionnaires, on obtient une valeur de  $z_j$  inférieure à  $d_{ij}$ . Complétons cette dernière observation par une comparaison des solutions optimales des  $(PL_1)$  et  $(PL_2)$ . Cette question a son importance puisque l'efficacité d'une énumération implicite dépend de la qualité de la borne utilisée.

Comme nous allons mettre en relation les variables des deux formulations dans ce qui suit, nous surmonterons d'un point les variables du  $(PL_1)$  lorsque l'on voudra leur faire référence, soit  $\hat{x}_{ij}$ ,  $\hat{y}_j$  et  $\hat{z}_j$ , et de deux points les variables du  $(PL_2)$ , soit  $\bar{y}_k$ . Ceci est nécessaire afin de lever toute ambiguïté puisque des variables de même

nom ont une signification différente dans chacune des formulations.

Il est aisé de vérifier par les propositions 1.8, 1.9 et 1.10 que les variables du  $(PL_1)$  et du  $(PL_2)$  satisfont les relations suivantes à l'optimalité en nombres entiers:

$$\dot{x}_{ij} = \sum_{k=1}^{|K_j|} a_{ik} \bar{y}_k \quad (1.14)$$

$$\dot{y}_j = \sum_{k=1}^{|K_j|} \bar{y}_k \quad (1.15)$$

On peut dès lors obtenir le résultat suivant.

**Proposition 1.11** *Toute solution réalisable du  $(PL_2)$  est aussi une solution réalisable du  $(PL_1)$ .*

**Preuve:** Soit une solution réalisable  $\bar{y}$  du  $(PL_2)$ . Donnons des valeurs aux variables  $\dot{x}_{ij}$  et  $\dot{y}_j$  du  $(PL_1)$  selon les équations (1.14) et (1.15). Par définition, nous avons

$$\sum_{j=1}^m \dot{x}_{ij} = \sum_{j=1}^m \sum_{k=1}^{|K_j|} a_{ik} \bar{y}_k = \sum_{k=1}^{\ell} a_{ik} \bar{y}_k \geq 1.$$

Il s'ensuit que (1.5) est donc toujours satisfaite. De la même façon, on montre que (1.7) est aussi toujours satisfaite. La contrainte (1.6) est nécessairement satisfaite puisque  $a_{ik} \bar{y}_k \leq \bar{y}_k$  où  $a_{ik} \in \{0, 1\}$ . Finalement, lorsque les  $\dot{x}_{ij}$  et  $\dot{y}_j$  sont fixées, la satisfaction de (1.4) est triviale.  $\square$

La conséquence principale de cette proposition est que les équations (1.14) et (1.15) peuvent servir de formules de passage pour transposer toute solution réalisable du  $(PL_2)$  vers  $(PL_1)$ . De plus, notons que (1.14) et (1.15) conservent le choix des

sites  $j$  pour l'ouverture des facilités tout comme l'affectation globale des usagers  $i$  à ces sites. Nous allons maintenant montrer que la solution du  $(PL_1)$  provenant d'une solution réalisable du  $(PL_2)$  suite à l'application de (1.14) et (1.15) n'est pas nécessairement de même valeur que la solution du  $(PL_2)$ .

Considérons une solution réalisable  $\tilde{y}$  du  $(PL_2)$ . Par définition, sa valeur  $z_{PL_2}$  est

$$z_{PL_2} = \sum_{k=1}^{\ell} c_k \tilde{y}_k = \sum_{j=1}^m \sum_{k=1}^{|K_j|} c_k \tilde{y}_k = \sum_{j=1}^m \sum_{k=1}^{|K_j|} \tilde{y}_k \max_{i \in I} \{a_{ik} d_{ij}\}.$$

Ce qui nous donne

$$z_{PL_2} = \sum_{j=1}^m \sum_{k=1}^{|K_j|} \max_{i \in I} \{a_{ik} \tilde{y}_k d_{ij}\}. \quad (1.16)$$

Transposons la solution  $\tilde{y}$  précédente dans le formalisme du  $(PL_1)$  en utilisant les formules de passage (1.14) et (1.15), et notons sa valeur  $z_{PL_1}^T$ . Pour obtenir  $z_{PL_1}^T$ , il suffit de résoudre

$$z_{PL_1}^T = \sum_{j=1}^m \dot{z}_j \quad \text{avec} \quad \dot{z}_j \geq d_{ij} \dot{x}_{ij} \quad i = 1, \dots, n; \quad j = 1, \dots, m$$

où  $\dot{x}_{ij}$  est donné par (1.14). Sous une écriture plus compacte, nous obtenons

$$z_{PL_1}^T = \sum_{j=1}^m \max_{i \in I} \{d_{ij} \dot{x}_{ij}\} = \sum_{j=1}^m \max_{i \in I} \left\{ d_{ij} \sum_{k=1}^{|K_j|} a_{ik} \tilde{y}_k \right\},$$

d'où l'équation

$$z_{PL_1}^T = \sum_{j=1}^m \max_{i \in I} \left\{ \sum_{k=1}^{|K_j|} a_{ik} \tilde{y}_k d_{ij} \right\}. \quad (1.17)$$

Maintenant, nous pouvons démontrer l'affirmation que nous avons annoncée précédemment.

**Proposition 1.12** *Soit  $z_{PL_2}$  la valeur d'une solution réalisable du  $(PL_2)$  et soit  $z_{PL_1}^T$  la valeur de cette solution transposée dans le formalisme du  $(PL_1)$  selon les formules de passage (1.14) et (1.15),  $z_{PL_1}^T$  et  $z_{PL_2}$  satisfont  $z_{PL_1}^T \leq z_{PL_2}$ .*

**Preuve:** Soit une solution réalisable  $\tilde{y}$  du  $(PL_2)$ . Choisissons un usager  $i^*$  tel que

$$i^* \in \arg\max_{i \in I} \left\{ \sum_{k=1}^{|K_j|} a_{ik} \tilde{y}_k d_{ij} \right\}$$

pour chacun des sites  $j$  de  $J$ . Alors, par le choix de  $i^*$ , on a

$$\max_{i \in I} \left\{ \sum_{k=1}^{|K_j|} a_{ik} \tilde{y}_k d_{ij} \right\} = \sum_{k=1}^{|K_j|} a_{i^*k} \tilde{y}_k d_{i^*j}.$$

De plus, étant donné que pour  $k = 1, \dots, |K_j|$ ,

$$a_{i^*k} \tilde{y}_k d_{i^*j} \leq \max_{i \in I} \{a_{ik} \tilde{y}_k d_{ij}\},$$

il s'ensuit alors que

$$\sum_{k=1}^{|K_j|} a_{i^*k} \tilde{y}_k d_{i^*j} \leq \sum_{k=1}^{|K_j|} \max_{i \in I} \{a_{ik} \tilde{y}_k d_{ij}\},$$

d'où

$$\max_{i \in I} \left\{ \sum_{k=1}^{|K_j|} a_{ik} \tilde{y}_k d_{ij} \right\} \leq \sum_{k=1}^{|K_j|} \max_{i \in I} \{a_{ik} \tilde{y}_k d_{ij}\}.$$

Si l'on fait la sommation sur l'ensemble des sites  $j$ , nous retombons sur les équations (1.16) et (1.17) et nous concluons alors que  $z_{PL_1}^T \leq z_{PL_2}$ .  $\square$

Nous avons égalité entre les valeurs  $z_{PL_1}^T$  et  $z_{PL_2}$  si et seulement si

$$\max_{i \in I} \left\{ \sum_{k=1}^{|K_j|} a_{ik} \tilde{y}_k d_{ij} \right\} = \sum_{k=1}^{|K_j|} \max_{i \in I} \{a_{ik} \tilde{y}_k d_{ij}\} \quad (1.18)$$

pour chaque site  $j$ . Puisque nous sommes dans un contexte de recouvrement, il existe alors un ensemble d'utilisateurs critiques différent pour chacune des colonnes  $k$  d'un  $K_j$  donné. Il s'ensuit que (1.18) est satisfait si et seulement si nous retenons pour une solution réalisable du  $(PL_2)$  au plus une colonne de coût non nul par sous-ensemble  $K_j$ . Sinon nous obtenons une stricte inégalité entre  $z_{PL_1}^T$  et  $z_{PL_2}$ .

Considérons maintenant la valeur optimale  $z_{PL_2}^*$  du  $(PL_2)$ . Par la proposition 1.11, nous savons que la solution optimale du  $(PL_2)$  transposée par (1.14) et (1.15) dans

le formalisme du  $(PL_1)$  est une solution réalisable. Soit  $z_{PL_1}^T$  la valeur de cette solution transposée. En combinant les propositions 1.11 et 1.12, nous obtenons  $z_{PL_2}^* \geq z_{PL_1}^T \geq z_{PL_1}^*$  où  $z_{PL_1}^*$  est la valeur optimale du  $(PL_1)$ . Ceci nous amène donc à l'étude de l'égalité des valeurs  $z_{PL_1}^*$  et  $z_{PL_2}^*$ . Afin d'apporter un début de réponse à cette question, exprimons l'inégalité entre  $z_{PL_1}^T$  et  $z_{PL_1}^*$  comme suit

$$z_{PL_1}^T = \sum_{j=1}^m \max_{i \in I} \{d_{ij} \dot{x}_{ij}\} \geq \sum_{j=1}^m \max_{i \in I} \{d_{ij} x_{ij}^*\} = z_{PL_1}^* \quad (1.19)$$

où  $\dot{x}_{ij}$  est donné par la formule de passage (1.14) et  $x_{ij}^*$  est la valeur optimale pour la variable  $x_{ij}$  du  $(PL_1)$ .

La nature de l'équation (1.19) ne nous permet pas de raisonner terme à terme puisque rien ne laisse supposer la dominance des termes de  $z_{PL_1}^T$  sur ceux de  $z_{PL_1}^*$ . Par contre, il est clair que si la formule (1.14) donnait à tout  $\dot{x}_{ij}$  la valeur  $x_{ij}^*$ , on aurait alors l'égalité entre  $z_{PL_1}^*$  et  $z_{PL_2}^*$ . Or ceci constitue une condition suffisante beaucoup trop forte (et donc aucunement nécessaire) étant donné qu'il suffit que la solution optimale du  $(PL_1)$  ne soit pas unique pour qu'elle puisse être violée même si  $z_{PL_1}^* = z_{PL_2}^*$ . Par contre, cette condition trop forte suppose implicitement que l'équation (1.18) soit satisfaite pour tout site  $j$ . Ainsi, nous pouvons assurer que l'égalité entre  $z_{PL_1}^*$  et  $z_{PL_2}^*$  est possible uniquement si la solution optimale du  $(PL_2)$  ne contient pas plus d'une colonne de coût non nul par  $K_j$ . Or ce qui était une condition nécessaire et suffisante pour satisfaire l'égalité des valeurs  $z_{PL_1}^T$  et  $z_{PL_2}^*$  n'est plus qu'une condition nécessaire dans le cas de  $z_{PL_1}^*$  et  $z_{PL_2}^*$  puisqu'une solution réalisable du  $(PL_2)$  possédant qu'une seule colonne de coût non nul par  $K_j$  n'est pas forcément optimale.

Bien que nous n'ayons rien de plus précis sur les conditions qu'il faut rencontrer pour satisfaire l'égalité entre  $z_{PL_1}^*$  et  $z_{PL_2}^*$ , il semble qu'une telle égalité soit assez exceptionnelle. Considérons l'exemple suivant pour s'en persuader.

**Exemple:** Soit l'instance de la figure 1.1 où  $a$  et  $b$  représentent deux usagers et  $A$ ,  $B$  et  $C$  trois sites potentiels pouvant accepter l'ouverture d'une facilité. La distance entre un usager et un site est indiquée par les nombres entre les flèches. Posons que le nombre maximal de facilités pouvant être ouvertes est de 2, c.-à-d. que  $p = 2$ .

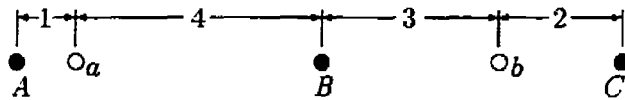


Figure 1.1 Exemple illustrant la qualité des bornes de  $(PL_1)$  et  $(PL_2)$

Par une énumération des cas, on constate alors que la seule solution optimale possible consiste à affecter l'utilisateur  $a$  au site  $A$  et l'utilisateur  $b$  en  $C$ , ce qui nous donne pour valeur optimale  $z^* = 3$ . La résolution du  $(PL_1)$  correspondant nous donne une solution en variables fractionnaires de valeur  $z_{PL_1}^* = 2,6154$  (où  $y_A = x_{aA} = 0,8205$ ,  $x_{bA} = 0,1026$ ,  $y_C = x_{bC} = 0,8974$ ,  $x_{aC} = 0,1795$ , et  $y_B = x_{aB} = x_{bB} = 0$ ) alors qu'avec  $(PL_2)$  on obtient directement la solution optimale en nombres entiers décrite précédemment.  $\square$

Ainsi, pour une instance donnée du (PCS) dont la solution optimale du  $(PL_2)$  est en nombres entiers, il faut donc s'attendre à obtenir pour  $(PL_1)$  une solution optimale fractionnaire. Quoiqu'il en soit, le théorème suivant découle directement de la combinaison des propositions 1.11 et 1.12 lorsqu'on les applique à une solution optimale du  $(PL_2)$ .

**Théorème 1.13** *Pour toute instance du  $p$ -Centre-Somme, la résolution de  $(PL_1)$  ne peut pas donner une meilleure borne inférieure que celle obtenue par  $(PL_2)$ .*

En dernière remarque, signalons que la solution optimale du  $(PL_2)$  en plus d'être une borne inférieure du  $(PCS)$  est aussi une borne supérieure du  $(PL_1)$ . On peut donc considérer  $(PL_1)$  comme une relaxation du  $(PL_2)$ . D'ailleurs puisque  $a_{ik} \in \{0, 1\}$ , la formule de passage (1.14) ne peut donner plus de  $2^{|K_j|}$  valeurs possibles à chacun des  $\dot{x}_{ij}$ . Or la solution optimale du  $(PL_1)$  implique que les valeurs optimales  $x_{ij}^*$  ne se limitent pas à l'ensemble des valeurs définies par (1.14). Ainsi,  $(PL_2)$  est effectivement plus restrictif que  $(PL_1)$ .

## 1.6 Propriétés spécifiques du $(PL_2)$

Étant donné la faible dégénérescence relative de  $(PL_2)$  ainsi que la qualité de sa valeur optimale comme borne inférieure comparativement à celle du  $(PL_1)$ , la formulation du  $(PCS_2)$  sera donc utilisée pour résoudre le problème du *p-Centre-Somme*. Ceci nous amène à considérer maintenant quelques propriétés se rapportant aux solutions optimales du  $(PL_2)$ . Ces dernières nous seront nécessaires pour discuter de la convergence de l'énumération implicite présentée au prochain chapitre.

**Proposition 1.14** *Soit une solution optimale du  $(PL_2)$  en nombres entiers, il existe alors au moins un usager critique qui est exclusif à chacune des colonnes de la solution.*

**Preuve:** Notons qu'une solution optimale du  $(PL_2)$  en nombres entiers est aussi solution optimale du  $(PCS_2)$ . Dès lors, la proposition 1.10 implique que dans une solution optimale du  $(PL_2)$  en nombres entiers il y a au plus une colonne choisie par site. Puisqu'une colonne choisie est un sous-ensemble  $I^*$ , la proposition 1.5 nous permet d'affirmer qu'il existe dans toute colonne de la solution au moins un usager

critique qui lui soit exclusif, c.-à-d., qui n'appartient à aucune autre colonne de la solution.  $\square$

Cette dernière proposition étant une condition nécessaire pour l'obtention d'une solution optimale du  $(PL_2)$  en nombres entiers, sa contraposée nous fournit alors une condition suffisante pour qu'une solution optimale du  $(PL_2)$  soit de valeur fractionnaire.

**Corollaire 1.15** *La présence dans une solution optimale du  $(PL_2)$  d'une colonne telle que chacun de ses usagers critiques appartient aussi à au moins une autre colonne de la solution implique que cette solution optimale soit fractionnaire.*

Notons qu'on désigne par solution fractionnaire toute solution du  $(PL_2)$  telle que certaines valeurs des variables  $y_k$  appartiennent à l'intervalle ouvert  $]0, 1[$ . Cet abus de langage provient de la littérature et il ne faut donc pas y voir une référence à la valeur de la fonction objectif.

Bien que le corollaire 1.15 nous donne une condition nécessaire pour l'obtention d'une solution optimale du  $(PL_2)$  fractionnaire, ce corollaire nous donne aucune information pouvant nous permettre d'identifier quelles doivent être les variables  $y_k$  de valeurs fractionnaires. La proposition suivante permet de conclure partiellement sur ce point.

**Proposition 1.16** *Toute valeur  $y_k^* \in ]0, 1[$  d'une solution optimale du  $(PL_2)$  implique que chacun des usagers critiques de la colonne  $k$  appartienne aussi à au moins une autre colonne de la solution.*



**Preuve:** Étant donné que  $y_k^* \in ]0, 1[$ , tous les usagers de la colonne  $k$  doivent appartenir aussi à au moins une autre colonne de la solution optimale afin de satisfaire les contraintes (1.11). La proposition est donc vérifiée puisque les usagers critiques d'une colonne appartiennent nécessairement à cette colonne.  $\square$

Or ceci ne nous garantit pas que toutes les colonnes d'une solution optimale telles que chacun des usagers critiques appartienne aussi à d'autres colonnes de la solution soient nécessairement de valeurs fractionnaires. Afin de montrer que de telles colonnes doivent toutes être de valeurs fractionnaires à l'optimum, nous allons associer à chacune des colonnes  $k$  d'une solution optimale du  $(PL_2)$  un ensemble  $\mathcal{K}(k)$  tel que  $\mathcal{K}(k)$  est constitué de toutes les colonnes de la solution, excepté la colonne  $k$  elle-même, qui contiennent au moins l'un des usagers critiques de  $k$ .

**Proposition 1.17** *Une solution optimale du  $(PL_2)$  fractionnaire implique que toute colonne  $k'$  telle que chacun de ses usagers critiques appartient à au moins l'une des colonnes  $k$  de l'ensemble  $\mathcal{K}(k')$  satisfait:*

1.  $y_{k'}^* \in ]0, 1[$ ;
2. il existe au moins une colonne  $k \in \mathcal{K}(k')$  telle que  $y_k^* \in ]0, 1[$ .

**Preuve:**

- (i) Montrons 1. Soit une solution optimale  $y^*$  du  $(PL_2)$  telle qu'il existe une colonne  $k'$  dont chacun de ses usagers critiques appartient aussi à au moins une colonne  $k$  de l'ensemble  $\mathcal{K}(k')$ . Posons que  $y_k^* \geq \varepsilon > 0$ ,  $\forall k \in \mathcal{K}$  et supposons

que  $y_{k'}^* = 1$ . Soit  $k' \in K_j$ . Si  $k'$  n'est pas la colonne de plus petit coût dans  $K_j$  (donc que tous les usagers de  $k'$  ne sont pas critiques), il existe alors dans  $K_j$  une colonne  $k''$  telle que  $k''$  est constituée de tous les usagers de  $k'$  privé de ses usagers critiques. Puisque  $y^*$  est optimale,  $y_{k''}^* = 0$  (toute autre valeur  $y_{k''}^*$  augmenterait le coût de  $y^*$ ).

Considérons la solution  $\hat{y}$  identique à  $y^*$  exceptée pour les colonnes  $k'$  et  $k''$  pour lesquelles  $\hat{y}_{k'} = 1 - \varepsilon$  et  $\hat{y}_{k''} = \varepsilon$  si la colonne  $k''$  existe. La solution  $\hat{y}$  est réalisable pour les contraintes (1.11) des usagers non critiques de  $k'$  ainsi que pour (1.12) puisque  $\hat{y}_{k'} + \hat{y}_{k''} = y_{k'}^*$ . Pour les usagers critiques de  $k'$ , les contraintes (1.11) sont aussi satisfaites dans  $\hat{y}$  puisque pour chacun de ces usagers,  $\hat{y}_{k'} + \sum_{k \in \mathcal{K}(k')} \hat{y}_k \geq 1$  est respectée.

Soient  $c_{k'}$  et  $c_{k''}$  les coûts respectifs des colonnes  $k'$  et  $k''$ , alors ce qui coûtait  $c_{k'}$  dans  $y^*$  coûte  $(1 - \varepsilon)c_{k'} + \varepsilon c_{k''}$  dans  $\hat{y}$ . Si la colonne  $k''$  n'existe pas dans  $K_j$ , le terme  $\varepsilon c_{k''}$  disparaît du coût de  $\hat{y}$  et on a  $c_{k'} > (1 - \varepsilon)c_{k'}$  puisque  $\varepsilon > 0$ . L'optimalité de  $y^*$  est ainsi contredite pour ce cas particulier. Si  $k''$  existe dans  $K_j$ , on peut alors réécrire le coût de  $k'$  comme étant le coût de l'affectation représentée par  $k''$  augmenté de ce qu'il faut payer pour se rendre jusqu'aux usagers critiques de  $k'$ , soit plus précisément  $c_{k'} = c_{k''} + \Delta c$  où  $\Delta c > 0$ . Puisque  $(1 - \varepsilon)c_{k'} + \varepsilon c_{k''} = c_{k'} - \varepsilon(c_{k''} + \Delta c) + \varepsilon c_{k''}$ , il s'ensuit que  $c_{k'} > c_{k'} - \varepsilon \Delta c$  après simplification. L'optimalité de  $y^*$  est aussi contredite dans le cas général et le point 1 de la proposition est donc démontré.

- (ii) Montrons 2. Soit la solution optimale  $y^*$  telle que  $y_{k'}^* \in ]0, 1[$ . Supposons que  $y_k^* = 1$  pour tout  $k \in \mathcal{K}(k')$ . Notons  $y_{k''}^*$  la valeur de la colonne  $k''$  dans  $y^*$  où  $k''$  est la colonne  $k'$  privée de ses usagers critiques. Considérons la solution  $\tilde{y}$  identique à  $y^*$  exceptée pour les colonnes  $k'$  et  $k''$  pour lesquelles  $\tilde{y}_{k'} = 0$  et  $\tilde{y}_{k''} \leftarrow y_{k'}^* + y_{k''}^*$ . La solution  $\tilde{y}$  est réalisable puisque les usagers critiques

de  $k'$  sont recouverts par les colonnes appartenant à  $\mathcal{K}(k')$  alors que la valeur donnée à  $k''$  satisfait les contraintes (1.11) des usagers non critiques de  $k'$  sans violer pour autant (1.12). Ainsi, ce qui coûtait  $y_{k'}^* c_{k'} + y_{k''}^* c_{k''}$  dans  $y^*$  coûte  $(y_{k'}^* + y_{k''}^*) c_{k''}$  dans  $\tilde{y}$ . Puisque  $c_{k'} > c_{k''}$ , la solution  $\tilde{y}$  a un coût moindre que  $y^*$  et l'optimalité de  $y^*$  est contredite. Il faut donc qu'au moins l'une des colonnes appartenant à  $\mathcal{K}(k')$  soit de valeur fractionnaire à l'optimum. Notons que si  $k''$  est vide, la solution  $\tilde{y}$  se réduit à retirer  $k'$  de  $y^*$  et l'on arrive à la même conclusion. Ainsi, la proposition est complètement démontrée.  $\square$

La combinaison des propositions 1.16 et 1.17 implique que dans toute solution optimale fractionnaire du  $(PL_2)$  seules les colonnes dont les usagers critiques appartiennent aussi à d'autres colonnes de la solution sont de valeurs fractionnaires. Toute autre colonne d'une telle solution prend une valeur entière puisqu'elle a au moins un usager critique qui lui est propre. De plus, la proposition 1.17 entraîne que le corollaire 1.15 constitue une condition nécessaire et suffisante pour qu'une solution optimale du  $(PL_2)$  soit fractionnaire. Étant donné la relation entre la proposition 1.14 et son corollaire 1.15, la proposition 1.14 est une condition nécessaire et suffisante pour qu'une solution optimale du  $(PL_2)$  soit en nombres entiers.

Notons que la proposition 1.16 s'applique aussi à la colonne  $k$  de valeur fractionnaire appartenant à  $\mathcal{K}(k')$  dans l'énoncé de la proposition 1.17. Ainsi, la proposition 1.17 s'étend au-delà de la démonstration de la nécessité et de la suffisance du corollaire 1.15 pour qu'une solution optimale du  $(PL_2)$  soit fractionnaire puisqu'elle nous informe qu'il existe toujours dans une telle solution au moins deux colonnes de valeurs fractionnaires et que celles-ci sont intimement liées.

Poursuivons avec un dernier résultat qui s'applique sans distinction aussi bien à une solution optimale fractionnaire qu'en nombres entiers.

**Proposition 1.18** *Sous l'hypothèse de la résolution du  $(PL_2)$  par l'algorithme du simplexe, il existe dans toute paire de colonnes d'une solution optimale du  $(PL_2)$  au moins un usager qui appartient qu'à une et une seule des colonnes de la paire.*

**Preuve:**

- (i) L'énoncé de la proposition est triviale si l'une des colonnes de la paire est de valeur entière puisque selon ce qui a été dit précédemment une telle colonne a au moins un de ses usagers critiques qui lui est propre que la solution optimale du  $(PL_2)$  soit fractionnaire ou en nombres entiers.
- (ii) Soit deux colonnes  $k_1$  et  $k_2$  d'une solution optimale  $y^*$  telle que  $y_{k_1}^*, y_{k_2}^* \in ]0, 1[$ . Si  $k_1$  et  $k_2$  ont au moins un usager qu'ils ne mettent pas en commun, la proposition est nécessairement respectée. Supposons donc que  $k_1$  et  $k_2$  soient constitués exactement des mêmes usagers. Or nous avons

$$c_{k_1} y_{k_1}^* + c_{k_2} y_{k_2}^* \geq (y_{k_1}^* + y_{k_2}^*) \min\{c_{k_1}, c_{k_2}\} \quad (1.20)$$

où  $c_{k_1}$  et  $c_{k_2}$  sont les coûts respectifs des colonnes  $k_1$  et  $k_2$ . Puisque l'inégalité stricte de (1.20) est obtenue pour  $c_{k_1} \neq c_{k_2}$ , nous avons alors avantage de retirer de  $y^*$  la colonne  $k_1$  ou  $k_2$  ayant le plus grand coût et d'ajouter sa valeur fractionnaire à celle de l'autre colonne de la paire afin de maintenir la satisfaisabilité des contraintes (1.11) et (1.12). On obtient ainsi une solution réalisable de coût moindre et l'optimalité de  $y^*$  est ainsi contredite. Pour le cas où  $c_{k_1} = c_{k_2}$ , il existe alors une infinité de solutions de même valeur que  $y^*$  et l'optimalité de  $y^*$  ne peut alors être contredite. Par contre, le domaine de ces solutions forme un

segment de droite continu dont les extrémités correspondent à prendre l'une ou l'autre des colonnes  $k_1$  et  $k_2$ , et de lui donner pour valeur fractionnaire  $y_{k_1}^* + y_{k_2}^*$ . Sachant que l'algorithme du simplexe résout un programme linéaire par une énumération de sommets du polytope formé par les contraintes du programme, nous pouvons affirmer qu'une solution telle que  $y^*$  ne peut être retenue suite à la résolution du  $(PL_2)$  par l'algorithme du simplexe puisque seules les solutions extrêmes du segment auquel appartient  $y^*$  seront considérées par l'algorithme. L'énoncé de la proposition est ainsi vérifié.  $\square$

Notons en terminant que si l'un des usagers critiques n'est pas mis en commun entre deux colonnes de valeurs fractionnaires, il doit alors exister dans la solution optimale du  $(PL_2)$  au moins une troisième colonne de valeur fractionnaire. Par contre, si aucun des usagers mis en commun dans une paire de colonnes fractionnaires est un usager critique, on ne peut être aussi catégorique sur la présence d'une autre colonne fractionnaire dans la solution puisque rien n'interdit *a priori* que ces usagers soient recouverts par une colonne de valeur entière. La proposition 1.18 ne nous permet donc pas d'assurer pour toute solution optimale du  $(PL_2)$  fractionnaire la présence d'une troisième colonne de valeur fractionnaire en plus des deux colonnes garantit par la proposition 1.17.

Or le point (i) de la démonstration de la proposition 1.17 a révélé un mécanisme qui semble donner du sens à l'existence de cette troisième colonne. Ainsi, une façon avantageuse de recouvrir des usagers non critiques d'une colonne de valeur fractionnaire qui ne sont pas partagés avec une autre colonne fractionnaire serait d'utiliser une colonne (ou plusieurs) contenant uniquement ces usagers. Puisque tous les usagers critiques d'une colonne fractionnaire appartiennent aussi à une autre colonne de la solution, il s'ensuit que la nouvelle colonne est nécessairement de valeur

fractionnaire (proposition 1.17).

Bien que ceci ne constitue pas une argumentation très rigoureuse, il reste qu'elle nous fournit une certaine compréhension sur la structure des solutions optimales fractionnaires du  $(PL_2)$ . De plus, signalons que les colonnes de valeurs fractionnaires d'une solution optimale doivent représenter des affectations d'utilisateurs à au moins deux sites  $j$  différents. Supposons que ce ne soit pas le cas (toutes les colonnes de valeurs fractionnaires appartiennent donc au même  $K_j$ ), la colonne fractionnaire de plus grand coût contredit alors la proposition 1.17 puisqu'elle ne peut pas partager ses utilisateurs critiques avec aucune autre colonne de valeur fractionnaire.

## CHAPITRE 2

### Résolution du problème

Nous allons maintenant suggérer une méthode de résolution exacte pour le problème du *p-Centre-Somme* sous sa formulation (PCS<sub>2</sub>). Cette méthode suit l'approche développée par Hansen, Jaumard et Poggi de Aragão (1991) pour résoudre les programmes mixtes.

L'approche de ces auteurs met en œuvre un algorithme de séparation et évaluation où chaque nœud des sous-problèmes est résolu par génération de colonnes. On peut trouver dans Chvátal (1983) et Lasdon (1970) des considérations théoriques d'ordre général sur la technique de génération de colonnes. Pour l'utilisation des méthodes de séparation et évaluation, consulter Nemhauser et Wolsey (1988), Parker et Rardin (1988) ainsi que Taha (1975).

#### 2.1 Justification de la génération de colonnes

Dans l'approche de la génération de colonnes, on désigne par la colonne  $k$  l'ensemble des informations contenues dans la  $k^{\text{ème}}$  colonne d'une matrice. Rappelons que la technique de la génération de colonnes consiste à introduire une colonne dans la matrice représentative du problème seulement lorsque l'information contenue dans cette colonne devient nécessaire pour la poursuite de la résolution du problème. Naturellement, il est alors sous-entendu que l'on sait construire implicitement chacune

des colonnes de la matrice. Puisque chaque variable  $y_k$  du  $(PL_2)$  représente un recouvrement d'utilisateurs par un cercle critique, il s'ensuit que la formulation du  $(PL_2)$  a une structure qui se prête bien à cette technique.

Bien que le  $(PL_2)$  ne contienne pas un nombre exponentiel de colonnes (il y en a au plus  $mn$ ), l'utilisation de la génération de colonnes est nécessaire puisque l'augmentation du nombre de colonnes lorsqu'on s'approche des instances de la réalité rend impossible une conservation explicite et complète de celles-ci dans la mémoire d'un ordinateur ou dépasse simplement la capacité des codes commerciaux de l'algorithme du simplexe. Prenons le cas type d'un réseau de téléphonie cellulaire pour une ville d'un million d'habitants. Pour une telle ville, on doit s'attendre à utiliser près de 150 cellules pour en recouvrir toute la superficie. Pour chacune de ces cellules, nous avons environ 3 sites potentiels pour l'installation d'une antenne, ce qui donne  $m = 450$ . Une cellule doit pouvoir desservir en moyenne jusqu'à 35 usagers simultanément, donc  $n = 5250$ . Ce problème réel nécessite donc un peu plus de 2,3 millions de variables. Naturellement, il est hors de question de résoudre un tel problème par une méthode où toutes les colonnes doivent être explicitées.

D'ailleurs, ce nombre important de variables dans les instances de la réalité rend impraticable une autre approche qui aurait pu se révéler intéressante autrement. Celle-ci consiste en l'utilisation des algorithmes conçus pour résoudre efficacement les problèmes de recouvrement en variables 0-1 (Balas, 1980; Balas et Ho, 1980; Fisher et Kedia, 1990; Harche et Thompson, 1994). La littérature concernant ces algorithmes mentionne des résolutions d'instances ne dépassant pas 10 000 variables et 800 contraintes. Outre la limitation de la taille des instances traitables par ces algorithmes, nous entrevoyons deux difficultés à résoudre. La première réside dans le fait que ces algorithmes ne tiennent compte d'aucune limitation dans le nombre de couvertures. Remarquons que cette difficulté n'est pas compliquante puisqu'on peut



toujours intégrer la limitation à même la règle de branchement de l'énumération implicite des algorithmes. La difficulté suivante concerne la densité des instances résolues dans la littérature. Elle ne dépasse pas les 10% alors qu'une instance du (PL<sub>2</sub>) a une densité supérieure à 50% (plus exactement, elle est de  $\frac{n+3}{2(n+1)}$  si on considère la formulation explicitant toutes les colonnes, soit de 50.019% pour  $n = 5250$ ). Nous devrions procéder à de nouvelles expérimentations afin d'estimer dans quelle mesure cette dernière difficulté est vraiment gênante et ainsi déterminer comment l'approche d'une résolution par les algorithmes de recouvrement serait prometteuse. Pour notre part, nous croyons que l'effort n'en vaut pas vraiment la peine étant donné la taille des instances que nous projetons résoudre.

## 2.2 Décomposition

Prenons une formulation matricielle telle que toutes les colonnes soient explicitement énumérées, la résolution du (PL<sub>2</sub>) se réduit alors à choisir les colonnes qui vont former la solution optimale. Puisque chaque colonne représente une affectation particulière d'utilisateurs à un site donné, le choix d'une colonne implique qu'une facilité doit être ouverte sur le site qui est associé à cette colonne. L'affectation des utilisateurs pour cette facilité doit alors être celle décrite par la colonne. On peut donc réduire (PL<sub>2</sub>) à un simple problème de localisation des facilités puisqu'une fois qu'une colonne est choisie, le site et l'affectation des utilisateurs à ce site sont déterminés. Ceci constitue ce que nous nommerons dorénavant le problème maître.

Plaçons-nous maintenant dans le contexte de la génération de colonnes, on n'explique donc plus toutes les colonnes de la matrice. Afin d'obtenir l'optimalité, il faudra

générer au besoin les colonnes nécessaires. Puisqu'une colonne consiste en une affectation d'utilisateurs à un site donné, nous allons donc nous définir un nouveau problème pour nous permettre de construire de *bonnes* affectations d'utilisateurs. La génération de colonnes en cours de résolution fait ressortir l'aspect affectation du (PCS<sub>2</sub>). Nous désignerons cet aspect comme étant le problème auxiliaire.

La technique de la génération de colonnes nous permet donc de séparer deux aspects du problème qui sont intimement liés, soit la localisation et l'affectation, en deux problèmes indépendants. Dès lors, la résolution du (PL<sub>2</sub>) dans un contexte de génération de colonnes se réduit à une suite de résolutions alternant entre le problème maître et le problème auxiliaire.

### 2.2.1 Problème maître

Le problème maître à une itération courante se déduit directement du (PL<sub>2</sub>). Comme toutes les colonnes sont implicitement connues (puisque l'on sait comment construire chacune d'elles), nous n'explicitons au départ de la résolution qu'un nombre restreint de colonnes. En fait, il faut que la matrice de départ soit au moins carrée et réalisable, en d'autres termes qu'elle soit constituée par au moins  $n + 1$  variables et qu'elle admette au moins une solution. Autrement, il faudrait recourir à des variables artificielles.

Définissons  $K' = \{1, \dots, k, \dots, \ell'\}$  avec  $m \gg \ell' \geq n + 1$  et  $K' \subseteq K$  comme étant l'ensemble des vecteurs-colonnes générés à l'itération courante. Nous obtenons alors

le programme suivant, noté (PM):

$$\min \quad \sum_{k=1}^{\ell'} c_k y_k \quad (2.1)$$

$$(PM) \quad \text{s.c.} \quad \sum_{k=1}^{\ell'} a_{ik} y_k \geq 1 \quad i = 1, 2, \dots, n \quad (2.2)$$

$$\sum_{k=1}^{\ell'} y_k \leq p \quad (2.3)$$

$$0 \leq y_k \leq 1 \quad k = 1, 2, \dots, \ell' \quad (2.4)$$

La définition des variables et des constantes ainsi que la signification des contraintes sont identiques à celles déjà données pour (PCS<sub>2</sub>).

### 2.2.2 Problème auxiliaire

Le but du problème auxiliaire est de fournir la meilleure colonne possible afin de poursuivre l'optimisation du (PM). Puisque nous allons résoudre le (PM) par l'algorithme du simplexe, il s'agit donc de trouver la colonne ayant le coût réduit le plus négatif. (Notons qu'en pratique la minimisation du coût réduit n'est pas une nécessité, il suffit qu'il soit négatif.) Ainsi, nous sommes assurés que la colonne qui sera ajoutée au (PM) entrera en base et que cette colonne amènera en même temps la plus grande diminution possible de la valeur objective. (De même, plusieurs colonnes de coût réduit négatif peuvent être introduites simultanément dans la matrice du (PM).) S'il est impossible de trouver une colonne ayant un coût réduit strictement négatif, alors (PM) est à l'optimum.

Comme le coût réduit d'une colonne à un site  $j$  est indépendant des affectations aux autres sites, nous allons donc résoudre un problème auxiliaire différent pour chaque site. On note  $(PA)_j$  le problème associé au site  $j$ .

$$\min \quad z_j - \sum_{i=1}^n a_i u_i - u_{n+1} \quad (2.5)$$

$$(PA)_j \quad \text{s.c.} \quad z_j \geq a_i d_{ij} \quad i = 1, 2, \dots, n \quad (2.6)$$

$$a_i \in \{0, 1\} \quad i = 1, 2, \dots, n \quad (2.7)$$

La variable  $a_i$  est égale à 1 si l'utilisateur  $i$  est affecté au site  $j$  et à 0 sinon. La variable  $z_j$  prend la valeur de la distance entre  $j$  et l'utilisateur  $i$  le plus éloigné qui lui est affecté.  $z_j$  constitue donc le coût de la colonne. Comme pour  $(PCS_1)$ ,  $z_j \geq 0$  est implicite pour tout  $j \in J$ . Le coefficient  $u_i$  prend la valeur de la variable duale associée à la  $i^{\text{ème}}$  contrainte (2.2) et la constante  $u_{n+1}$ , celle de la variable duale associée à la contrainte (2.3). Les valeurs de ces variables duales proviennent de la dernière optimisation du (PM). De plus, à cause du sens des inégalités des contraintes (2.2) et (2.3),  $u_i \geq 0$  pour tout  $i \in I$  et  $u_{n+1} \leq 0$ . L'objectif (2.5) correspond au coût réduit de la colonne. Les contraintes (2.6) constituent des contraintes de seuil et (2.7) imposent qu'un usager soit complètement ou pas du tout affecté en  $j$ .

## 2.3 Propriétés et résolution du problème auxiliaire

Chaque  $(PA)_j$  a une structure particulière telle que lorsqu'on l'exploite, sa résolution devient très facile. Ayant recours à des listes ordonnées, Hansen, Labbé et Minoux (1994) ont proposé une méthode de résolution exacte dont la complexité est polynomiale.

### 2.3.1 Listes ordonnées et recouvrements d'utilisateurs

Soit  $\mathcal{L} = \{x_1, x_2, \dots, x_t\}$  où  $t$  est un entier, nous dirons que  $\mathcal{L}$  est une liste ordonnée d'éléments  $x$  selon une fonction  $f(\cdot)$  si et seulement si pour tout  $1 \leq r < s \leq t$  nous avons  $f(x_r) \leq f(x_s)$ . Nous supposons qu'il existe un ordre lexicographique selon lequel les éléments  $x$  puissent être ordonnés d'une manière unique. Ainsi, en présence de deux éléments  $x$  de même valeur, soit  $f(x_r) = f(x_s)$  où  $r \neq s$ , l'ordre lexicographique prévaudra alors pour ordonner ces deux éléments. De cette façon, nous obtiendrons toujours une seule et unique liste  $\mathcal{L}$  pour un ensemble donné d'éléments  $x$ .

Pour chaque site  $j$ , donnons-nous une liste ordonnée  $\mathcal{L}$  des utilisateurs  $i$  selon la distance séparant chacun des  $i$  à  $j$ . Cette liste est notée  $L_j$ . Formellement, nous avons donc  $L_j = \{i_1, i_2, \dots, i_n\}$  où  $d_{i_1 j} \leq d_{i_2 j} \leq \dots \leq d_{i_n j}$ . Nous allons maintenant montrer que les listes  $L_j$  contiennent suffisamment d'informations pour énumérer tous les recouvrements d'utilisateurs en chacun des sites  $j$ .

Considérons pour un site  $j$  les affectations données par la méthode suivante. Prenons l'utilisateur le plus proche de  $j$ , soit  $i_1$  seul, afin de constituer la première affectation  $\{i_1\}$ . Construisons ensuite une deuxième affectation en utilisant les deux premiers

usagers de  $L_j$ , ce qui donne l'affectation  $\{i_1, i_2\}$ . On poursuit avec les trois premiers usagers, puis les quatre et ainsi de suite jusqu'à la dernière affectation possible, soit  $\{i_1, i_2, i_3, i_4, \dots, i_n\}$ . Notons que cette dernière affectation correspond à  $L_j$ . Si l'on reprend la méthode sur chacun des sites, ceci nous donne exactement  $mn$  affectations. Étant donné que ces affectations sont aussi des listes ordonnées et qu'elles sont nécessairement complètes (la présence d'un usager dans une affectation en  $j$  implique que tous les usagers qui le précèdent dans  $L_j$  soient aussi dans l'affectation), nous allons donc les désigner dorénavant sous le terme de sous-liste complète de  $L_j$ .

Notons que tout recouvrement d'usagers par un cercle critique centré au site  $j$  peut s'écrire comme une liste ordonnée d'usagers selon leur distance à  $j$ . Or il est toujours possible d'ordonner les usagers d'une telle liste pour qu'elle corresponde à l'une des sous-listes complètes de  $L_j$ . Puisque toutes les sous-listes complètes de  $L_j$  sont strictement différentes les unes des autres et qu'à un recouvrement correspond une seule et unique sous-liste complète de  $L_j$ , il s'ensuit que la méthode d'énumération des sous-listes complètes que nous avons décrite précédemment peut aussi être utilisée pour énumérer tous les recouvrements possibles d'usagers au site  $j$ .

Nous allons maintenant montrer que les sous-listes complètes de  $L_j$  constituent un puissant outil pour résoudre  $(PA)_j$ . Ceci est d'autant plus vrai qu'on obtient sans grand effort, soit en  $O(mn \log n)$  avec un algorithme de tri, toutes les listes  $L_j$ . De plus, nous avons à déployer cet effort qu'une seule fois, juste avant de débiter la résolution d'une instance du  $(PCS_2)$ .

### 2.3.2 Méthode de résolution

Nous savons par définition qu'une colonne du (PM) doit correspondre à un ensemble d'utilisateurs recouverts par un cercle critique. Puisque l'utilité du problème auxiliaire est de fournir des colonnes au (PM), la solution optimale du (PA)<sub>j</sub> doit nécessairement représenter un recouvrement d'utilisateurs au site  $j$ . Sachant qu'un recouvrement d'utilisateurs en  $j$  peut s'écrire comme une sous-liste complète de  $L_j$ , on pourrait alors considérer une énumération des sous-listes complètes de  $L_j$  et retenir comme solution du (PA)<sub>j</sub> la sous-liste dont le coût réduit serait le plus petit. Or selon la proposition 1.4, nous avons au plus  $mn$  cercles critiques possibles. Ceci implique que parmi les  $mn$  sous-listes complètes qui seront énumérées sur l'ensemble des sites  $j$ , il est possible qu'un certain nombre de sous-listes ne corresponde pas à un recouvrement d'utilisateurs.

Lorsqu'un utilisateur est ajouté à une sous-liste complète afin de produire une nouvelle sous-liste complète, on s'attend à ce que le rayon de l'affectation représentée par la première sous-liste soit augmenté jusqu'à ce qu'un nouveau cercle critique passe par l'utilisateur qui a été ajouté. Ceci est effectivement vrai quand l'utilisateur que l'on ajoute est strictement plus éloigné du centre  $j$  du cercle que son prédécesseur immédiat dans  $L_j$ . Autrement, si l'utilisateur est à la même distance du centre que son prédécesseur, la première sous-liste ne peut donc pas correspondre à un recouvrement puisqu'il n'y aura aucune augmentation du rayon du cercle critique suite à l'ajout de cet utilisateur. Nous dirons alors qu'une sous-liste complète de  $L_j$  est maximale si et seulement si il y a une stricte augmentation du rayon lorsqu'on passe à la sous-liste complète suivante. Les sous-listes complètes maximales de  $L_j$  sont donc les seules sous-listes représentatives d'un recouvrement d'utilisateurs au site  $j$ .

La question est donc de savoir si la résolution d'un  $(PA)_j$  par une énumération des sous-listes complètes de  $L_j$  peut générer une colonne inacceptable pour le (PM). Selon la proposition suivante, la réponse est négative.

**Proposition 2.1** *Il existe une sous-liste complète de  $L_j$  qui optimise  $(PA)_j$  et qui correspond nécessairement à un recouvrement d'utilisateurs au site  $j$ .*

**Preuve:** Soit  $L_1^j$  une sous-liste de  $L_j$ . Pour tout utilisateur  $i \in L_1^j$  nous avons  $a_i = 1$  et pour tout autre  $i$ ,  $a_i = 0$ . Le coût  $z_1$  de l'affectation donnée par  $L_1^j$  a pour valeur  $\max_{i \in L_1^j} \{a_i d_{ij}\}$ . Considérons les deux cas suivants:

- (i) Posons que la sous-liste  $L_1^j$  ne soit pas complète et que son dernier utilisateur soit  $i_1$ . Considérons  $L_2^j$ , la sous-liste complète de  $L_j$  ayant aussi  $i_1$  pour dernier utilisateur. Le coût associé à  $L_2^j$  est donc  $z_1$ . Puisque  $L_2^j$  contient plus d'utilisateurs que  $L_1^j$  et que  $u_i \geq 0$  pour tout  $i \in I$ , il s'ensuit que la somme des  $a_i u_i$  de  $L_2^j$  dans l'objectif (2.5) sera supérieure ou égale à celle de  $L_1^j$ . Les sous-listes complètes de  $L_j$  sont donc des solutions réalisables du  $(PA)_j$  qui dominent (mais pas nécessairement au sens strict) les solutions données par toutes les autres sous-listes de  $L_j$ . Ainsi, si  $L_1^j$  est une solution optimale du  $(PA)_j$  alors  $L_2^j$  doit l'être aussi.
- (ii) Posons que la sous-liste  $L_1^j$  est complète mais non maximale. Ajoutons un utilisateur à  $L_1^j$  afin de construire la sous-liste complète suivante puis un autre et ainsi de suite jusqu'à ce qu'on obtienne une sous-liste complète maximale, notée  $L_2^j$ . Puisque  $L_2^j$  a au moins un utilisateur de plus que  $L_1^j$  pour le même coût  $z_1$  et que tout  $u_i \geq 0$ , il s'ensuit que les sous-listes complètes maximales de  $L_j$  dominent aussi toutes les autres sous-listes de  $L_j$  de sorte que si  $L_1^j$  optimise  $(PA)_j$ , il en est de même pour  $L_2^j$ .



Il doit donc exister une sous-liste complète maximale de  $L_j$  qui soit une solution optimale du  $(PA)_j$ .  $\square$

Ainsi, une résolution des  $(PA)_j$  par une énumération des sous-listes complètes en chacun des sites  $j$  est possible. De plus, il n'est pas nécessaire d'évaluer systématiquement le coût réduit associé à chacune des sous-listes complètes. En fait, ce calcul est pertinent seulement pour les sous-listes complètes maximales. Une façon rapide de s'assurer qu'une sous-liste complète est maximale, c'est-à-dire qu'elle corresponde à un recouvrement d'utilisateurs, consiste à vérifier la stricte augmentation du rayon de l'affectation (ou de son coût) si l'utilisateur suivant de  $L_j$  est ajouté à la sous-liste. L'Algorithme 1 qui suit donne la procédure à suivre pour résoudre le problème auxiliaire au site  $j$ .

### Algorithme 1

Considérons  $\tau$  comme étant l'indice se référant à l'utilisateur  $i$  qui occupe la  $\tau^{\text{ième}}$  position dans  $L_j$ . Prenons  $L$  pour désigner la sous-liste complète de  $L_j$  à l'itération courante et  $\mu$ , la valeur de la sommation des variables duales associées avec les utilisateurs appartenant à  $L$ . Les variables duales  $u_i$  pour tout  $i \in I$  et  $u_{n+1}$  sont considérées comme des constantes déjà fixées.

#### 1. Initialisation.

Poser  $L \leftarrow \emptyset$  (la sous-liste vide de  $L_j$ ),  $\tau \leftarrow 0$ ,  $\mu \leftarrow u_{n+1}$  et  $\bar{c}^* \leftarrow \infty$ .

#### 2. Test d'arrêt.

Si  $\tau = n$ , alors arrêter.  $L^*$  est la sous-liste complète optimale du  $(PA)_j$  et le

coût réduit qui lui est associé vaut  $\bar{c}^*$ .

Sinon, poser  $\tau \leftarrow \tau + 1$ . Ajouter ensuite le nouvel usager  $i_\tau$  à la suite de  $L$  et poser  $\mu \leftarrow \mu + u_{i_\tau}$ .

### 3. Validation de la sous-liste courante.

Si  $L$  n'est pas une sous-liste complète maximale de  $L_j$ , retourner à l'étape 2.

### 4. Calcul du coût réduit courant.

Évaluer  $\bar{c} \leftarrow d_{i_\tau j} - \mu$  où  $\bar{c}$  représente le coût réduit associé à la sous-liste  $L$  courante et  $d_{i_\tau j}$ , la distance entre le dernier usager ajouté à  $L$  et le site  $j$ .

### 5. Conservation de la meilleure sous-liste.

Si  $\bar{c} < \bar{c}^*$ , alors poser  $L^* \leftarrow L$  et  $\bar{c}^* \leftarrow \bar{c}$ . Poursuivre à l'étape 2.

Le résultat suivant garantit la convergence de l'Algorithme 1 vers l'optimalité ainsi que sa complexité.

**Proposition 2.2** *L'Algorithme 1 résout exactement  $(PA)_j$  en  $O(n)$ .*

**Preuve:** Puisque toutes les sous-listes complètes de  $L_j$  sont construites une par une à l'étape 2 et que seulement celles qui sont maximales ont accès à l'étape 5, il s'ensuit qu'une sous-liste satisfaisant la proposition 2.1 sera nécessairement retenue comme solution optimale du  $(PA)_j$ . Étant donné que chacune des  $n$  sous-listes complètes de  $L_j$  n'est considérée qu'une seule fois et que l'évaluation du coût réduit d'une sous-liste se calcule toujours à partir du coût réduit de la sous-liste précédente (une opération qui s'effectue en  $O(1)$ ), il en résulte que la complexité de l'Algorithme 1 est de  $O(n)$ . □

## 2.4 Résolution de la relaxation linéaire

On constate aisément que l'Algorithme 1 est insuffisant pour résoudre d'un seul coup le problème auxiliaire. Par contre, cet algorithme nous fournit une sous-liste complète de  $L_j$  optimale pour chacun des  $(PA)_j$ . Puisque nous cherchons à générer la meilleure colonne à introduire dans la matrice du (PM), cette colonne doit nécessairement correspondre à la sous-liste appartenant à l'ensemble des sous-listes optimales obtenues par la résolution de chacun des  $(PA)_j$  dont le coût réduit est le plus petit. Partant de ce fait, il suffit donc d'appliquer l'Algorithme 1 sur chacun des  $L_j$  et de conserver la sous-liste optimale de coût réduit minimal pour résoudre le problème auxiliaire. Comme il y a  $m$  sites potentiels, la complexité totale d'une itération de génération de colonnes est  $O(mn)$  (proposition 2.2).

Une fois le problème auxiliaire résolu, il reste à transcrire la sous-liste solution sous la forme d'un vecteur-colonne. Sous sa forme générique, une colonne  $k$  de la matrice du (PM) est constituée de la façon suivante:

$$\begin{bmatrix} c_k \\ a_{1k} \\ a_{2k} \\ \vdots \\ a_{nk} \\ 1 \end{bmatrix}$$

où  $c_k$  est le coût de la colonne qu'on retrouve dans l'objectif (2.1), les  $a_{1k}, a_{2k}, \dots, a_{nk}$  constituent les coefficients des contraintes (2.2) et le 1 au bas de la colonne correspond au coefficient dans (2.3). Ainsi, pour générer une colonne  $k$  à partir d'une sous-liste complète de  $L_j$ , il suffit de poser  $a_{ik} = 1$  pour tout usager  $i$  de la sous-liste et  $a_{ik} = 0$  pour les autres usagers. Le coût  $c_k$  de  $k$  est déterminé par la distance du dernier usager de la sous-liste de  $L_j$ .

Naturellement, on suppose que la colonne  $k$  que l'on vient de générer a un coût réduit strictement négatif. Si ce n'est pas le cas, il est alors inutile d'ajouter  $k$  à la matrice du (PM) puisque l'algorithme du simplexe n'introduira pas celle-ci en base. On sait dès lors que (PM) est à l'optimum. Ceci signifie que toutes les colonnes nécessaires à la résolution ont été générées et que par le fait même, la solution optimale du (PM) coïncide avec celle du (PL<sub>2</sub>). Nous avons donc intérêt à ajouter  $k$  et à poursuivre la résolution du (PM) dans le cas contraire.

Formellement, ce qui précède donne l'Algorithme 2.

## Algorithme 2

Prenons  $K'$  pour désigner l'ensemble des colonnes de la matrice du (PM) à l'itération courante. Définissons la colonne  $k_j$  comme étant la colonne générée à partir de la sous-liste optimale obtenue par l'application de l'Algorithme 1 sur la liste  $L_j$ . Le coût réduit associé à  $k_j$  est  $\bar{c}_j$ . Notons par  $k$  la colonne de coût réduit  $\bar{c}$  tel que  $\bar{c} = \min_{j \in J} \bar{c}_j$ .

### 1. Initialisation.

Poser  $K'$  tel que  $K' \subseteq K$  et  $mn \gg |K'| \geq n + 1$ . De plus,  $K'$  doit admettre pour le (PM) au moins une solution réalisable.

### 2. Évaluation des variables duales.

Par l'algorithme du simplexe résoudre à l'optimum le (PM) défini par l'ensemble  $K'$  courant. On obtient ainsi les valeurs des variables duales  $u_i$  pour tout  $i \in I$  et  $u_{n+1}$  qui seront utilisées par l'Algorithme 1.

### 3. Génération d'une colonne.

#### 3.1 Initialisation.

Posons  $j \leftarrow 0$  et  $\bar{c} \leftarrow \infty$ .

#### 3.2 Test d'arrêt.

Si  $j = m$ , alors passer à l'étape 4. Sinon, poser  $j \leftarrow j + 1$ .

#### 3.3 Résolution du $(PA)_j$ .

Appliquer l'Algorithme 1 sur  $L_j$  et en déduire la colonne  $k_j$ .

#### 3.4 Conservation de la meilleure colonne.

Si  $\bar{c}_j < \bar{c}$ , alors poser  $k \leftarrow k_j$  et  $\bar{c} \leftarrow \bar{c}_j$ . Retourner à l'étape 3.2.

### 4. Test d'optimalité.

Si  $\bar{c} \geq 0$ , alors arrêter. Il n'existe pas de colonne  $k$  qui puisse être ajoutée à  $K'$  telle que la valeur objectif du (PM) puisse encore diminuer éventuellement.

Sinon, poser  $K' \leftarrow K' \cup \{k\}$ . Poursuivre à l'étape 2.

On peut considérer l'Algorithme 2 comme étant une variante de l'algorithme du simplexe. En fait, les étapes 3 et 4 que nous avons introduites ont pour seule utilité de déterminer qu'elle est la prochaine colonne à entrer en base et de l'ajouter aux colonnes qui sont hors base sans se préoccuper d'effectuer le pivot puisqu'il sera réalisé par l'algorithme du simplexe à l'étape 2. Il s'ensuit alors que les résultats relatifs à l'algorithme du simplexe s'étendent à l'Algorithme 2. En conséquence, nous avons ce qui suit.

**Proposition 2.3** *Sous l'hypothèse de non-dégénérescence à chaque itération, l'Algorithme 2 résout exactement  $(PL_2)$  en un nombre fini d'itérations.*

**Preuve:** Consulter le théorème 6 de la section 1.2.1 de Lasdon (1970).  $\square$

Lorsqu'il y a dégénérescence, la valeur de l'objectif ne décroît plus et il y a alors possibilité que l'algorithme se mette à cycler sur un ensemble de changements de base. Afin d'assurer la convergence de l'algorithme du simplexe dans ce cas, il faut recourir à une règle d'anticyclage. Or l'algorithme du simplexe ne cycle pas en pratique même en présence de la dégénérescence (à moins de construire un problème spécifique en ce sens). De plus, la règle d'anticyclage est coûteuse en temps de calcul. Ainsi, il est rare qu'un code commercial de l'algorithme du simplexe utilise la règle d'anticyclage lors de la résolution d'un programme linéaire. Par contre, il a souvent recours à une méthode de perturbation de la base dégénérée non pas pour s'assurer de la convergence de l'algorithme, mais plutôt pour atténuer les effets de la dégénérescence sur la résolution.

L'utilisation d'une règle d'anticyclage à l'étape 2 ne serait pas suffisante pour assurer la convergence de l'Algorithme 2 puisqu'elle ne peut pas tenir compte des colonnes qui n'ont pas encore été générées. Afin de compléter cette dernière règle, on pourrait vérifier si la colonne  $k$  proposée par l'étape 3 de l'Algorithme 2 ne se trouve pas déjà dans  $K'$  avant d'y être insérée. Si c'était le cas, il faudrait alors générer une autre colonne sinon l'Algorithme 2 cyclerait sur une suite de colonnes. Une façon simple mais coûteuse d'introduire une règle d'anticyclage sur la génération de colonnes serait de modifier l'étape 5 de l'Algorithme 1 comme suit:

##### 5. Conservation de la meilleure sous-liste.

Si  $\bar{c} < \bar{c}^*$  et si  $L$  correspond à aucune colonne  $k \in K'$ ,  
alors poser  $L^* \leftarrow L$  et  $\bar{c}^* \leftarrow \bar{c}$ . Poursuivre à l'étape 2.

D'autre part, étant donné que l'algorithme du simplexe ne cycle pas en pratique, le vecteur des valeurs duales résultant de l'étape 2 de l'Algorithme 2 devrait donc être différent d'une itération à l'autre et ne jamais se répéter. Ainsi, une colonne ne devrait pas être générée plus d'une fois durant la résolution du (PM). En conséquence, on peut s'attendre à ce que l'Algorithme 2 ne cycle pas plus que l'algorithme du simplexe dans la pratique. Pour cette raison, nous avons choisi de ne pas implanter la règle d'anticyclage relative à la génération de colonnes puisque cette protection n'apporterait rien de plus et ralentirait inutilement l'algorithme. Naturellement, la règle serait de mise si jamais un tel cyclage était observé.

Maintenant que nous avons une méthode efficace pour résoudre la relaxation linéaire du (PCS<sub>2</sub>), abordons l'énumération implicite afin d'obtenir l'optimalité en nombres entiers.

## 2.5 Suggestion d'une énumération implicite

Nous proposons d'effectuer une énumération implicite selon l'approche primale de la méthode de séparation et évaluation décrite dans Hansen, Jaumard et Poggi de Aragão (1991). En peu de mots, une méthode de séparation et d'évaluation consiste à séparer le domaine de solutions en sous-domaines jusqu'à ce que chacun des sous-domaines amène une solution en nombres entiers. Dès lors, on trouve la solution optimale par comparaison. En pratique, la séparation du domaine s'obtient par le développement d'un arbre, nommé arbre de branchement, où chaque nœud de l'arbre est associé à un sous-domaine du domaine de solutions. Chacune des séparations est effectuée selon une règle de branchement et nous dirons que l'on passe d'un nœud père à un nœud fils lorsqu'un domaine (le père) est séparé en sous-domaines (les fils)

et que l'un de ceux-ci est évalué.

Étant dans un contexte de génération de colonnes, nous pouvons définir la règle de branchement de manière à brancher soit sur le problème maître, soit sur le problème auxiliaire. Il existe plusieurs arguments en faveur d'un branchement sur le problème auxiliaire qu'on peut trouver dans Barnhart *et al.* (1994). Mentionnons seulement qu'un branchement sur le problème maître amène souvent une destruction de la structure du problème auxiliaire en y introduisant de nouvelles variables duales alors qu'un branchement sur le problème auxiliaire produit plutôt de nouvelles dépendances entre les variables de ce dernier, ce qui est moins destructeur surtout si on peut éviter de les expliciter en les incorporant, par exemple, à même les structures de données. Pour le  $(PCS_2)$ , un branchement sur le problème auxiliaire revient à interdire la génération de certaines affectations d'utilisateurs.

Désignons par  $K_r$  l'ensemble réduit des colonnes non vides permises à un nœud donné de l'arbre de branchement tel que  $K_r \subseteq K$ . Soit  $K'$ , l'ensemble des colonnes qui sont explicites et générées à l'itération courante d'une évaluation. Alors sous la condition que l'Algorithme 1 puisse générer que des colonnes appartenant au  $K_r$  représentatif du sous-domaine de solutions, l'Algorithme 2 peut être utilisé pour évaluer le (PM) au nœud correspondant puisque  $K' \subseteq K_r$ . Bien que ceci constitue l'essentiel de l'évaluation, il reste à définir une règle de branchement ainsi que la manière de l'appliquer et de développer l'arbre de branchement afin d'obtenir efficacement la solution optimale du  $(PCS_2)$ . Nous supposons donc pour l'instant que l'on sait résoudre en nombres entiers le problème auxiliaire sous tout branchement. Nous reviendrons sur la résolution des  $(PA)_j$  sous le branchement à la section 2.6.



### 2.5.1 Règle de branchement

Pour résoudre une partition de tâches entre des exécutants, Ryan et Foster (1981) ont proposé une règle qui leur permettait d'obtenir des arbres de branchement plus équilibrés que la dichotomie habituelle qui impose  $y_k = 0$  sur une branche et  $y_k = 1$  sur l'autre où  $y_k$  est une variable du problème maître telle que  $y_k \in ]0, 1[$ . La règle de ces auteurs se formalise comme suit: soient  $t_1$  et  $t_2$  deux tâches distinctes, alors  $t_1$  et  $t_2$  sont affectées à un même exécutant ou bien,  $t_1$  et  $t_2$  sont affectées à deux exécutants différents.

Considérons maintenant deux usagers  $i_1$  et  $i_2$  distincts. Une règle de branchement du même type que celle de Ryan et Foster se formuleraient comme suit pour le (PCS<sub>2</sub>):

1. soit que les usagers  $i_1$  et  $i_2$  doivent être affectés à des facilités différentes, ce qui s'obtient par l'inégalité

$$a_{i_1} + a_{i_2} \leq 1; \quad (2.8)$$

2. soit que les usagers  $i_1$  et  $i_2$  doivent être affectés à la même facilité, d'où l'égalité

$$a_{i_1} = a_{i_2}. \quad (2.9)$$

Les branchements (2.8) et (2.9) sont tout indiqués pour un branchement sur le problème auxiliaire étant donné que  $a_{i_1}$  et  $a_{i_2}$  sont des variables déjà définies dans les  $(PA)_j$ , ce qui implique que  $a_{i_1}$  et  $a_{i_2} \in \{0, 1\}$ . Comme les nouvelles dépendances amenées par (2.8) et (2.9) entre les usagers  $i_1$  et  $i_2$  sont aucunement reliées aux sites  $j$ , les  $(PA)_j$  restent alors disjoints et leur résolution demeure indépendante. Naturellement, ceci nécessite que tout branchement (2.8) ou (2.9) doit être ajouté à chacun des  $(PA)_j$ .

Notons que les branchements (2.8) et (2.9) permettent que la colonne solution de tout (PA)<sub>j</sub> puisse contenir aucun des usagers  $i_1$  et  $i_2$ . La génération de telles colonnes est nécessaire, sinon on obtiendrait d'un côté du branchement des solutions optimales dont toutes les colonnes contiendraient l'un des deux usagers et de l'autre côté, des solutions optimales où les deux usagers appartiendraient à toutes les colonnes.

### 2.5.2 Sélection des usagers

Puisque nous voulons une énumération efficace, nous avons donc intérêt de bien choisir les deux usagers sur lesquels appliquer la règle de branchement. Dès lors, par le choix des usagers  $i_1$  et  $i_2$ , nous voulons interdire de chaque côté du branchement au moins une colonne de valeur fractionnaire. De plus, les branchements (2.8) et (2.9) appliqués sur  $i_1$  et  $i_2$  ne doivent pas interdire les mêmes colonnes. Ces exigences demandent donc que l'on prenne pour  $i_1$  et  $i_2$  deux usagers qui se retrouvent ensemble dans certaines colonnes fractionnaires et qui s'excluent mutuellement dans d'autres colonnes de valeurs fractionnaires. Ainsi, un branchement (2.8) interdirait les colonnes où  $i_1$  et  $i_2$  sont ensemble alors que (2.9) interdirait plutôt les colonnes ne contenant qu'un seul des usagers  $i_1$  et  $i_2$ .

D'autre part, nous avons vu à la section 1.6 qu'il suffit que chacune des affectations d'une solution optimale du (PL<sub>2</sub>) possède au moins un usager critique qui n'appartient à aucune autre affectation de la solution pour que la solution soit en nombres entiers. Comme les branchements (2.8) et (2.9) amènent une partition de l'ensemble des éléments sur lesquels ils sont appliquées, nous avons donc avantage de brancher sur des usagers critiques appartenant à des affectations de valeurs fractionnaires afin de forcer leur partitionnement et permettre ainsi l'émergence de la condition nécessaire d'obtention de solutions optimales en nombres entiers (proposition

1.14). Or la possibilité d'un tel choix d'utilisateurs tout en satisfaisant un branchement efficace n'est que partiellement garanti.

**Proposition 2.4** *Pour tout nœud de l'arbre de branchement ayant une solution optimale du (PM) fractionnaire, il existe dans cette solution au moins deux colonnes  $k_1$  et  $k_2$  de valeur optimale  $y_{k_1}^*$  et  $y_{k_2}^* \in ]0, 1[$  ainsi que deux utilisateurs  $i'$  et  $i''$  tels que  $i'$  est un utilisateur critique de  $k_1$  appartenant aussi à  $k_2$  et  $i''$  est soit un utilisateur de  $k_1$ , soit un utilisateur de  $k_2$ .*

**Preuve:** Soit une solution optimale fractionnaire du  $(PL_2)$ . Selon la proposition 1.17, il doit exister dans cette solution une colonne  $k_1$  telle que l'un des utilisateurs critiques de  $k_1$ , notons-le  $i'$ , appartient aussi à une autre colonne  $k_2$  de la solution et que nécessairement  $y_{k_1}^*$  et  $y_{k_2}^* \in ]0, 1[$ . Étant donné que  $(PL_2)$  est résolu par l'algorithme du simplexe, la proposition 1.18 assure qu'il existe alors dans cette dernière solution un utilisateur  $i''$  distinct de  $i'$  n'appartenant qu'à une seule des deux colonnes  $k_1$  et  $k_2$ . Comme chaque ensemble  $K_r$  associé aux nœuds de l'arbre de branchement est un ensemble de colonnes représentant des affectations possibles d'utilisateurs ( $K_r \subseteq K$ ), ce que nous venons de déduire par les propositions 1.17 et 1.18 pour  $(PL_2)$  est aussi valide pour toute solution optimale fractionnaire du (PM) à tout nœud de l'arbre de branchement.  $\square$

Si on choisit  $i'$  comme utilisateur  $i_1$  et  $i''$  comme utilisateur  $i_2$ , la proposition 2.4 n'assure aucunement que  $i_2$  puisse être un utilisateur critique des colonnes  $k_1$  ou  $k_2$ . Par contre, le choix de  $i_1$  et  $i_2$  tel que proposé par cette dernière proposition satisfait les exigences du branchement efficace discutées plus haut bien qu'il ne permet pas toujours un branchement exclusivement orienté vers l'émergence de la condition nécessaire

apportant une solution optimale en nombres entiers. En conséquence, la sélection des usagers consistera à prendre, s'ils existent, les usagers  $i_1$  et  $i_2$  comme étant respectivement des usagers critiques des colonnes de valeurs fractionnaires  $k_1$  et  $k_2$  telles que  $i_1 \in k_2$  et  $i_2 \notin k_1$ . (Notons que prendre  $i_2$  comme usager critique de  $k_1$  plutôt que  $k_2$  avec  $i_2 \notin k_2$  serait moins pratique étant donné que toute colonne a au moins un usager critique mais pas nécessairement deux.) Si une telle paire d'usagers n'existe pas, on prendra alors toute autre paire d'usagers satisfaisant la proposition 2.4. Ainsi, on branchera toujours au moins sur un usager critique.

### 2.5.3 Convergence de l'énumération

Le théorème 5.1 de Parker et Rardin (1988) garantit la convergence de la méthode classique (décrite par leur Algorithme 5A) de séparation et évaluation pour tout programme linéaire en variables 0-1. De part la généralité de l'Algorithme 5A, la formulation du théorème 5.1 suppose que la séparation satisfasse les 3 conditions suivantes:

1. seul un nombre fini de sous-problèmes peut être généré quelle que soit la séquence des séparations du domaine de solutions;
2. suite à toute séparation, le sous-problème de chacun des nœuds fils est strictement différent du domaine de solutions du père;
3. le domaine de solutions du père est toujours égale à l'union des sous-domaines de ses fils.

L'application récursive des conditions 2 et 3 sur l'ensemble des nœuds de l'arbre de branchement implique alors que la séparation ne génère jamais plus d'une fois un

sous-problème donné et qu'elle ne soustrait aucune partie du domaine de solutions à l'évaluation. Outre ces 3 conditions, la preuve de convergence de l'Algorithme 5A ne nécessite aucune autre hypothèse.

Il est facile de vérifier que l'énumération implicite que nous suggérons est un cas particulier de l'Algorithme 5A. La démonstration de sa convergence se réduit donc à montrer que la séparation satisfait les 3 conditions du théorème 5.1 de Parker et Rardin. Ainsi, nous pouvons formaliser le résultat qui suit.

**Proposition 2.5** *L'application de la règle de branchement telle que définie par (2.8) et (2.9) sur deux usagers  $i_1$  et  $i_2$  satisfaisant la proposition 2.4 assure la convergence de la méthode de séparation et évaluation pour toute instance du problème du p-Centre-Somme en un nombre fini de branchements.*

**Preuve:**

- (i) Montrons que la condition 1 est satisfaite. Considérons le nombre de branchements (2.8) ou (2.9) nécessaire pour fixer toutes les variables  $a_i$  de  $(PA)_j$  en ne posant qu'une seule de celles-ci à 1 (soit le cas où  $(PA)_j$  n'a plus qu'une solution réalisable unique suite à de tels branchements), il s'ensuit qu'on ne peut pas accumuler plus de  $\frac{n}{2}(n-1)$  branchements (2.8) et plus de  $n-1$  branchements (2.9) à tout moment de l'énumération. Le nombre de nœuds dans l'arbre de branchement est par conséquent borné par le produit de ces deux quantités (ce qui nous donne  $O(n^3)$ ). La condition 1 est donc satisfaite et l'énumération finie.
- (ii) Montrons que les conditions 2 et 3 sont satisfaites. Considérons deux colonnes  $k_1$  et  $k_2$  de valeurs fractionnaires d'une solution optimale du (PM) à un nœud

donné de l'arbre de branchement telles que  $k_1$  et  $k_2$  possèdent deux usagers  $i_1$  et  $i_2$  tels que  $i_1 \in k_1$ ,  $i_2 \notin k_1$  et  $i_1, i_2 \in k_2$  (la proposition 2.4 assure l'existence d'une telle paire de colonnes). Comme  $a_{i_1}$  et  $a_{i_2} \in \{0, 1\}$ , un branchement (2.8) sur  $i_1$  et  $i_2$  interdit la colonne  $k_2$  au nœud fils correspondant ainsi que toutes les colonnes contenant  $i_1$  et  $i_2$  alors que le branchement (2.9) interdit  $k_1$  et toutes les autres colonnes ayant qu'un seul des usagers  $i_1$  et  $i_2$  au nœud fils auquel il est associé. Il est donc impossible qu'une solution réalisable du (PM) puisse être commune aux deux sous-domaines résultants d'une telle séparation puisque par définition, les usagers  $i_1$  et  $i_2$  doivent être affectés à au moins une facilité. On obtient donc une bipartition du domaine de solutions sans qu'aucune des solutions réalisables en nombres entiers du (PM) soit exclue des fils, d'où la satisfaction de la condition 3. Comme la solution optimale du père ne satisfait ni le branchement (2.8) ni (2.9), le sous-problème correspondant ne peut donc pas être régénéré chez ses fils et il en découle que la condition 2 est satisfaite.  $\square$

Discutons maintenant de la qualité de l'énumération implicite suggérée par la proposition 2.5.

#### 2.5.4 Qualité du branchement

La bipartition du domaine de solutions au nœud père entraîne pour chaque fils une solution optimale distincte strictement différente de celle du père. De ce fait, tous les branchements sont ainsi pertinents et apportent de l'information immédiatement utile à la recherche de solutions en nombres entiers. Il s'ensuit que la sélection des

usagers satisfaisant la proposition 2.4 rencontre largement les exigences du branchement efficace discutées à la section 2.5.2. Or un tel choix d'usagers n'est pas nécessaire à l'obtention d'un branchement équilibré car du moment que les usagers  $i_1$  et  $i_2$  amènent la bipartition du domaine de solutions au nœud père et que le nombre de nœuds reste fini, la convergence de l'énumération implicite est assurée même si ni  $i_1$  ni  $i_2$  est un usager critique d'une des colonnes de la solution optimale.

Par contre, interdire une colonne par l'un de ses usagers critiques permet son remplacement par une colonne qui diffère très peu de la colonne interdite (pensons aux colonnes de même centre et de moindre coût que la colonne interdite). On peut donc s'attendre à ce que la résolution du (PM) aux nœuds fils en soit facilitée étant donné que le remplacement de la colonne interdite n'implique pas de grandes modifications à la solution du père. Le développement de l'arbre de branchement permettrait ainsi de conserver d'un nœud à l'autre ce qui a été acquis précédemment. Il faut s'attendre en retour à un arbre de branchement plus profond qu'avec un branchement sur des usagers non critiques. Puisqu'un tel branchement interdirait alors plus de colonnes que nécessaires, nous aurions d'importantes modifications à apporter à la solution optimale du (PM) en passant d'un nœud père à un nœud fils. On perdrait ainsi les acquis des solutions antérieures et permettrait du même coup à la dégénérescence de se manifester aisément en dehors de la racine de l'arbre de branchement.

Notons finalement que l'énumération telle que décrite par la proposition 2.5 n'assure pas la stricte augmentation des valeurs optimales du (PM) entre un père et ses fils bien que chacun des fils a une solution optimale distincte de celle du père. Ceci est dû au fait que rien n'interdit d'obtenir à un nœud père donné une solution fractionnaire dont la valeur est celle de l'optimum du (PCS<sub>2</sub>).

### 2.5.5 Rapprochement avec les bornes supérieures généralisées

À tout nœud de l'arbre de branchement, nous avons  $K = K_r \cup K_o$  où  $K_o$  est l'ensemble des colonnes interdites au nœud courant. Soit une séparation du sous-domaine du nœud courant par l'application des branchements (2.8) et (2.9) sur le problème auxiliaire. Nous obtenons alors les relations suivantes:

$$K_r = K_r^G \cup K_o^{(2.8)} = K_r^D \cup K_o^{(2.9)}$$

où  $K_r^G$  et  $K_r^D$  sont respectivement les ensembles réduits des colonnes permises au fils gauche et au fils droit alors que  $K_o^{(2.8)}$  et  $K_o^{(2.9)}$  sont respectivement les ensembles des nouvelles colonnes interdites par les branchements (2.8) et (2.9).

Supposons que toutes les colonnes de  $K_r$  sont explicitées dans la matrice du (PM), on pourrait dès lors obtenir les mêmes ensembles  $K_r^G$  et  $K_r^D$  en introduisant directement au problème maître une contrainte du type

$$\sum_{k \in K_o^{(x)}} y_k \leq 0 \quad (2.10)$$

où  $(x) = \{(2.8), (2.9)\}$ . Puisque par définition  $y_k \geq 0$  pour tout  $k \in K$ , le branchement (2.10) implique alors que  $y_k = 0$  pour tout  $k \in K_o^{(x)}$  (ce qui correspond à interdire ces dernières colonnes au nœud fils).

Or le branchement (2.10) a été introduit par Beale et Tomlin (1970) sous le nom de *division selon un ensemble ordonné de variables* pour la résolution de problèmes de localisation non convexe sous des contraintes disjointes du type  $\sum \lambda_i = 1$  avec  $\lambda_i \in \{0, 1\}$  où  $i$  est l'indice de la variable  $\lambda$ . La littérature désigne plutôt ce branchement comme étant une *division dichotomique sur une borne supérieure généralisée* (*GUB dichotomy* en anglais).



Il s'ensuit donc que le branchement sur le problème auxiliaire tel que défini par (2.8) et (2.9) est une application de la division dichotomique sur une borne supérieure généralisée dans un contexte où toutes les variables ne sont pas explicitées. Un tel branchement a du sens pour le  $(PSC_2)$  puisque la contrainte (1.12) n'est rien d'autre qu'une borne supérieure générale sur le nombre de facilités pouvant être ouvertes même si les colonnes constituant  $K_o^{(x)}$  ne correspondent pas à l'usage habituel de (2.10). Ainsi, tout ce qui est connu sur ce type de branchement (voir Nemhauser et Wolsey, 1988) est aussi valide pour un branchement sur le problème auxiliaire tel que nous le pratiquons.

### 2.5.6 Stratégie de branchement

Comme l'arbre de branchement devrait être assez profond, nous proposons une exploration de l'arbre de branchement par une recherche en profondeur d'abord pure (recherche qui consiste, par exemple, à toujours brancher en premier sur le fils gauche). Puisque la borne supérieure sera donnée par la première solution en nombres entiers rencontrée lors de l'énumération, nous voulons que celle-ci soit obtenue rapidement et qu'elle puisse être la meilleure possible.

Comme on peut accumuler au plus  $\frac{n}{2}(n-1)$  branchements (2.8) et  $n-1$  branchements (2.9) (point (i) de la preuve de la proposition 2.5), on pourrait croire qu'il est avantageux de brancher en premier avec (2.9). Or des expériences préliminaires ont montré qu'en branchant plutôt avec (2.8), l'énumération implicite nécessitait en général beaucoup moins de nœuds; ceci est dû au fait que la borne supérieure ainsi obtenue se situait dans l'arbre de branchement presque toujours au voisinage de la solution optimale du  $(PCS_2)$ . On branchera donc avec (2.8) en premier.

### 2.5.7 Modification de la matrice du (PM)

Selon l'étape 1 de l'Algorithme 2, l'ensemble  $K'$  doit admettre une première solution réalisable suite à tout branchement. Ainsi, il est exclu de retirer physiquement de  $K'$  les colonnes ne satisfaisant pas un branchement puisque la réalisabilité de  $K'$  ne pourrait pas être garantie bien que  $K' \subseteq K_r$ . Nous allons donc pénaliser les colonnes non réalisables de  $K'$  avec un coût suffisamment élevé pour les obliger à sortir de la base lors de la résolution du (PM) sur le domaine du nœud fils. Si une colonne pénalisée reste en base à l'optimalité, nous saurons que le (PM) est devenu irréalisable au nœud courant. D'autre part, nous n'avons pas besoin de retirer la pénalisation en remontant l'arbre de branchement puisque les informations relatives aux branchements seront retirées des (PA)<sub>j</sub> au fur et à mesure de la remontée. Ainsi, l'étape 3 de l'Algorithme 2 pourra régénérer au besoin les colonnes pénalisées permises à nouveau.

## 2.6 Résolution du problème auxiliaire sous le branchement

L'Algorithme 1 tel que présenté à la section 2.3 ne permet pas l'application de l'énumération implicite décrite à la section 2.5 puisqu'on ne peut pas garantir que toutes les colonnes pouvant être générées par cet algorithme appartiennent nécessairement à  $K_r$  où  $K_r$  est l'ensemble des colonnes représentatif d'un sous-domaine de solutions à un nœud donné de l'arbre de branchement. Il faut donc apporter des modifications à l'Algorithme 1 afin de satisfaire les exigences de l'énumération.

### 2.6.1 Problème auxiliaire sous un branchement général

Prenons  $\mathcal{I}^{\leq}$  pour représenter l'ensemble des associations d'utilisateurs deux par deux générées par le branchement (2.8) et  $\mathcal{I}^=$  pour l'ensemble des associations deux par deux résultant du branchement (2.9). Posons que plusieurs branchements ont été effectués aussi bien avec un branchement qu'avec l'autre afin d'obtenir une forme la plus générale du  $(PA)_j$ . Ainsi, la forme générique du  $(PA)_j$  sous un branchement général, notée  $(PAG)_j$ , est donnée par le programme suivant:

$$\min \quad z_j - \sum_{i=1}^n a_i u_i - u_{n+1} \quad (2.11)$$

$$\text{s.c.} \quad z_j \geq a_i d_{ij} \quad i = 1, 2, \dots, n \quad (2.12)$$

$$(PAG)_j \quad a_{i_1} + a_{i_2} \leq 1 \quad \forall \{i_1, i_2\} \in \mathcal{I}^{\leq} \quad (2.13)$$

$$a_{i_1} = a_{i_2} \quad \forall \{i_1, i_2\} \in \mathcal{I}^= \quad (2.14)$$

$$a_i \in \{0, 1\} \quad i = 1, 2, \dots, n \quad (2.15)$$

La définition des variables et des constantes ainsi que les explications relatives à la fonction objectif et aux contraintes de seuil restent identiques à celles déjà données pour  $(PA)_j$ .

Les contraintes (2.14) peuvent être éliminées par un changement de variables et nommons  $(PAG)_j'$  le problème résultant d'un tel changement. Sachant que les contraintes (2.13) et (2.15) définissent le polytope du problème du *Stable Pondéré Maximal* sur un graphe (Hammer, Hansen et Simeone, 1982; Nemhauser et Trotter, 1975), on

pourrait alors résoudre  $(\text{PAG})'_j$  par l'algorithme de Bron et Kerbosch (Christofides pp.33-35, 1975) ou de Trushin (1980) à la condition que les contraintes (2.12) soient dualisées. Puisque le problème du *Stable Pondéré Maximal* est reconnu dans la littérature comme étant un problème NP-dur, nous aurions alors avantage de dualiser (2.13) afin de retrouver un problème de la structure du  $(\text{PA})_j$  que l'on sait résoudre en temps polynomial par l'Algorithme 1 (proposition 2.2). Or on obtient ainsi une relaxation lagrangienne du  $(\text{PAG})'_j$ , ce qui implique que la génération de colonnes reste malgré tout non polynomiale puisque la détermination des multiplicateurs de Lagrange est coûteuse en pratique et qu'une énumération implicite doit être prévue afin d'assurer la réalisabilité de la colonne proposée par la résolution de la relaxation.

Nous avons vu à la section 2.3 que les contraintes (2.12) amenaient une structure très forte en donnant un ordre dans lequel affecter les usagers à un site potentiel, structure qui est détruite par la dualisation de (2.12) ou (2.13). Nous allons montrer qu'il est possible de conserver cette structure sous le branchement en généralisant l'interprétation qu'il faut donner aux listes  $L_j$ . Ainsi,  $(\text{PAG})_j$  pourra être résolu aussi facilement que  $(\text{PA})_j$ .

### 2.6.2 Méthode de résolution

Considérons une liste  $L_j$  telle que  $L_j = \{i_1, i_2, \dots, i_n\}$  et posons que chaque élément appartenant à  $L_j$  est un usager critique potentiel pour une affectation d'usagers au site  $j$ . Soit  $E_j = (e_1^j, \dots, e_k^j, \dots, e_n^j)$ , un vecteur-ligne de dimension  $n$  où  $e_k^j$  est une étiquette sur l'utilisateur  $i_k$  de  $L_j$  telle que  $e_k^j = 0$  si  $i_k$  est un usager critique d'une des affectations permises en  $j$  et que  $e_k^j > 0$  dans le cas contraire.

Soient deux usagers  $i_1$  et  $i_2$  tels que  $i_1$  précède  $i_2$  dans  $L_j$  ( $d_{i_1j} \leq d_{i_2j}$ ). Posons qu'un branchement (2.8) est appliqué sur les usagers  $i_1$  et  $i_2$ . Il s'ensuit alors que  $i_1$  et  $i_2$  doivent être affectés à des facilités différentes et que par conséquent, toute affectation contenant  $i_1$  et  $i_2$  est interdite. Puisque nous sommes dans un contexte de recouvrement, aucun des usagers de  $L_j$  se trouvant à une distance plus grande ou égale du site  $j$  que  $i_2$  ne peut être affecté en  $j$ . Ceci implique que ces usagers ne sont plus des usagers critiques potentiels pour toute affectation au site  $j$ . Ainsi, poser  $e_k^j \leftarrow e_k^j + 1$  pour tout  $k$  tel que  $d_{ikj} \geq d_{i_2j}$  permet de maintenir la cohérence de  $E_j$  avec (2.8).

D'autre part, l'application du branchement (2.9) sur  $i_1$  et  $i_2$  oblige que ces deux usagers soient affectés à la même facilité. Ainsi, les affectations contenant  $i_1$  sans  $i_2$  sont nécessairement interdites. Dès lors, tout usager dont la distance au site  $j$  est plus grande ou égale à celle de  $i_1$  mais plus petite que  $i_2$  ne peut être considéré comme un usager critique potentiel. La conservation de la cohérence entre  $E_j$  et (2.9) entraîne alors qu'on pose  $e_k^j \leftarrow e_k^j + 1$  pour tout  $k$  tel que  $d_{i_1j} \leq d_{ikj} < d_{i_2j}$ .

Afin de bien fixer les idées, nous allons maintenant illustrer par un exemple le maintien de la cohérence des vecteurs  $E_j$  sur une suite de branchements.

**Exemple:** Soient  $L_{j_1} = \{i_1, i_2, i_3, i_4, i_5, i_6, i_7\}$  et  $L_{j_2} = \{i_6, i_3, i_7, i_2, i_4, i_1, i_5\}$  telles que tous les usagers contenus dans  $L_{j_1}$  ou  $L_{j_2}$  sont à une distance différente du site à laquelle est associée la liste. Comme aucun branchement n'a encore été effectué, nous avons donc  $E_{j_1} = E_{j_2} = (0, 0, 0, 0, 0, 0, 0)$ .

Posons qu'un premier branchement du type (2.9) est appliqué sur  $i_3$  et  $i_4$  (c.-à-d. que  $a_{i_3} = a_{i_4}$ ), ce qui donne  $E_{j_1} = (0, 0, 1, 0, 0, 0, 0)$  et  $E_{j_2} = (0, 1, 1, 1, 0, 0, 0)$ .

Appliquons ensuite un branchement du type (2.8) sur  $i_2$  et  $i_6$  (soit  $a_{i_2} + a_{i_6} \leq 1$ ). On obtient ainsi  $E_{j_1} = (0, 0, 1, 0, 0, 1, 1)$  et  $E_{j_2} = (0, 1, 1, 2, 1, 1, 1)$ .  $\square$

L'accumulation des deux branchements de l'exemple impliquent qu'il ne reste plus que 4 affectations réalisables au site  $j_1$  (soit celles qui ont pour usagers critiques  $i_1, i_2, i_4, i_5$ ) et qu'une seule au site  $j_2$  (qui consiste en l'affectation unique de  $i_6$ ). Toutes autres affectations aux sites  $j_1$  et  $j_2$  ne peuvent donc pas satisfaire les contraintes induites par les branchements. Pour résoudre  $(\text{PAG})_{j_1}$  et  $(\text{PAG})_{j_2}$ , il suffit donc d'appliquer l'Algorithme 1 sur les cinq affectations décrites plus haut. Remarquons que la proposition 2.1 reste valide même sous le branchement.

De même, il faut aussi maintenir la cohérence des  $E_j$  lorsqu'on remonte dans l'arbre de branchement. Les usagers interdits comme usager critique à un site  $j$  doivent donc retrouver leur potentialité au nœud où ils l'ont perdue. Le retrait d'un branchement à chacun des  $(\text{PAG})_j$  s'obtient en opérant  $e_k^j \leftarrow e_k^j - 1$  pour chaque usager  $i_k$  affecté par le branchement au site  $j$ .

Les vecteurs  $E_j$  ont donc la capacité de rendre compte des branchements ainsi que de leur accumulation. Il n'est donc pas nécessaire d'introduire explicitement les branchements dans  $(\text{PAG})_j$  puisqu'il suffit de vérifier la valeur des étiquettes des usagers critiques pour savoir si une colonne est interdite ou non à un nœud de l'arbre de branchement.

### Algorithme 1 généralisé

Il suffit d'apporter la modification suivante à l'Algorithme 1 pour obtenir l'Algorith-

me 1 généralisé.

### 3. Validation de la sous-liste courante.

Si  $L$  n'est pas une sous-liste complète maximale de  $L_j$  ou si  $e_\tau^j > 0$ , retourner à l'étape 2.

Comme la vérification de la valeur de  $e_\tau^j$  s'effectue en  $O(1)$ , la complexité de l'Algorithme 1 généralisé est donc  $O(n)$  (proposition 2.2). Naturellement, il ne faut pas négliger la mise à jour des  $E_j$ . Ceci est possible en visitant une seule fois chaque usager de chaque liste  $L_j$ , soit en  $O(mn)$ . Il suffit de parcourir  $L_j$  à l'aide de deux pointeurs dont l'un sert à retenir la position de l'usager définissant la dernière sous-liste complète maximale rencontrée ainsi qu'à incrémenter ou à décrémenter l'étiquette  $e_k^j$  s'il y a lieu lors de son déplacement alors que l'autre pointeur recherche la position des usagers  $i_1$  et  $i_2$  sur lesquels le branchement s'applique.

## CHAPITRE 3

### Résultats

Nous allons présenter quelques résultats sur les conditions de résolution de la méthode décrite au chapitre 2. Les instances utilisées pour nos expérimentations correspondent à des cas particuliers de la formulation (PCS<sub>2</sub>). Nos résultats sont comparés avec ceux que Hansen, Labbé et Minoux ont obtenus avec la méthode que ces auteurs ont proposée en 1994. Leurs résultats sont d'ailleurs les seuls disponibles dans la littérature.

#### 3.1 Bancs d'expérimentation

La méthode de résolution présentée au chapitre 2 a été testée sur 37 types d'instances différentes classées suivant les paramètres  $m$ ,  $n$ ,  $\delta$  et  $\rho$  où  $m$  et  $n$  désignent respectivement la cardinalité des ensembles  $I$  et  $J$ ,  $\delta$  la dimension de l'espace dans laquelle se situe l'instance (soit le nombre de coordonnées nécessaires pour localiser dans cet espace chaque élément appartenant à  $I$  et à  $J$ ) et  $\rho$  le degré d'intersection entre les ensembles  $I$  et  $J$ , c.-à-d. le rapport du nombre d'utilisateurs qui sont aussi des sites potentiels pour l'ouverture d'une facilité sur le nombre total d'utilisateurs à desservir. Ces quatre paramètres traduisent donc des considérations d'ordre topologique. Chacune des 37 classes ainsi définie est constituée de 10 instances générées par un procédé aléatoire. Le générateur d'instances  $\mathcal{G}(m, n, \delta, \rho)$  utilisé détermine chaque élément des ensembles  $I$  et  $J$  d'une instance donnée en effectuant autant de fois que nécessaire



un tirage avec remise dans l'hypercube  $[0, 1000]^d$ . Comme le tirage suit la loi de distribution uniforme, chaque point de l'hypercube a la même probabilité d'être choisi à chacun des tirages. Le degré d'intersection  $\rho$  indique le nombre de tirages communs à  $I$  et à  $J$ .

Le générateur  $\mathcal{G}(m, n, \delta, \rho)$  a été choisi afin de permettre la comparaison des performances de notre méthode de résolution avec celle proposée par Hansen, Labbé et Minoux (1994). Signalons que ces derniers auteurs rapportent des expérimentations de résolution avec leur algorithme seulement pour des instances du (PCS) appartenant au carré  $[0, 1000]^2$  et telles que  $m = n$  et  $\rho = 1$ . Nous reprenons leur étude et l'étendons à des classes d'instances de degré d'intersection moindre ainsi que de dimensions supérieures à 2 (ceci en vue d'une extension du (PCS) à la classification). De plus, pour les besoins de la comparaison, nous avons adopté la norme euclidienne pour évaluer les distances entre les usagers et les sites potentiels.

Comme pour Hansen, Labbé et Minoux, les résultats présentés dans les tableaux qui vont suivre sont des valeurs moyennes obtenues à partir de la résolution des 10 instances de chaque classe. Les 10 instances de chacune des 37 classes ont été générées qu'une seule fois. Ainsi, nous avons toujours utilisé dans nos expérimentations les mêmes 10 instances pour une caractérisation topologique donnée. Ceci donne l'avantage de mieux comparer les conditions de résolution pour différentes valeurs de  $p$ .

La méthode de résolution présentée au chapitre 2 a été programmée en C++ et l'algorithme du simplexe de la librairie CPLEX 2.1 a été utilisé pour résoudre le (PM). Afin d'amorcer l'Algorithme 2, nous avons formé la matrice  $K'$  de départ en prenant dans un premier temps la colonne pleine (colonne ne contenant que des 1) de chaque ensemble  $K_j$ , puis les colonnes ayant un seul usager (colonnes formées

de deux 1 en comptant le coefficient de la contrainte (2.3) du (PM)), puis celles ayant deux usagers et ainsi de suite jusqu'à ce que  $|K'| \geq n + 1$ . Les colonnes sont ainsi introduites en bloc dans  $K'$  selon le nombre de 1 qu'elles contiennent. La réalisabilité de  $K'$  est alors assuré par la présence des colonnes pleines.

Les expérimentations ont été réalisées sur une station SPARC SUN 20/514MP équipée de 128 Mo de mémoire vive et gérée par le système d'exploitation Solaris 2.4-27. À notre connaissance, une station SS 20/514MP est approximativement quatre fois plus rapide que le VAX 8650 utilisé par Hansen, Labbé et Minoux pour leurs expérimentations.

### 3.2 Comparaison expérimentale des bornes

Par le théorème 1.13, nous savons que la borne inférieure donnée par la résolution du  $(PL_1)$  ne peut pas être meilleure que celle obtenue par  $(PL_2)$ . Or ce résultat théorique ne permet pas de prévoir si la différence entre ces deux bornes est significative ou non en pratique et quel en est l'ordre de grandeur.

Les colonnes  $PCS_1$  et  $PCS_2$  du tableau 3.1 donnent respectivement les sauts de dualité observés en moyenne pour la résolution d'une instance du (PCS) avec les formulations  $(PCS_1)$  et  $(PCS_2)$ . Nous définissons le saut de dualité comme étant le rapport entre la différence des valeurs des solutions optimales en nombres entiers et de la relaxation linéaire sur la valeur de la solution optimale en nombres entiers. Puisque les instances utilisées par Hansen, Labbé et Minoux en 1994 ne sont pas disponibles, nous avons donc généré nos propres instances avec  $\mathcal{G}(m, n, \delta, \rho)$  en prenant les mêmes valeurs que ces derniers auteurs (soit  $m = n$ ,  $\delta = 2$  et  $\rho = 1$ ).

Tableau 3.1 Comparaison du saut de dualité des formulations ( $PCS_1$ ) et ( $PCS_2$ )

$n$	$p$	$PCS_1$ (%)	$PCS_2$ (%)	$PCS_2$ -HLM (%)
10	2	33,9	4,0	4,0
	3	43,3	8,1	8,0
	5	52,7	12,4	10,0
20	3	36,8	2,7	2,0
	5	45,8	7,0	8,0
	10	57,9	17,3	14,0
30	5	41,5	4,3	3,0 <sup>†</sup>
	10	51,6	9,4	11,0
	15	58,7	15,3	11,0
40	5	41,2	1,5	3,0
	10	50,4	5,6	7,0
	15	56,2	9,4	11,0
50	10	46,8	4,9	5,0
	20	56,7	13,2	11,0

† valeur corrigée (le tableau original indiquait 30%).

Nous reportons dans la colonne  $PCS_2$ -HLM les sauts de dualité observés en moyenne par ces auteurs sur leurs instances. La comparaison des colonnes  $PCS_2$  et  $PCS_2$ -HLM montre que nos instances sont de même difficulté que celles de Hansen, Labbé et Minoux.

L'étude des colonnes  $PCS_1$  et  $PCS_2$  montre que la différence entre les valeurs optimales du  $(PL_1)$  et du  $(PL_2)$  est plus que significative. Ainsi, les conditions nécessaires qu'il faut rencontrer pour obtenir l'égalité entre les deux formulations sont bien aussi difficiles à rencontrer en pratique que le laissait croire l'exemple de la section 1.5 et qu'il ne faut donc pas s'attendre à les rencontrer.

D'autre part, nous confirmons la dégénérescence du  $(PL_1)$  telle que discutée à la section 1.4. Effectivement, nous avons observé une manifestation très importante du phénomène de dégénérescence lors de la résolution du  $(PL_1)$  par l'algorithme du simplexe au point de rendre instable les mécanismes de perturbation de CPLEX 2.1. De plus, la dégénérescence prend des proportions si importantes avec  $(PL_1)$

qu'il ne faut guère espérer pouvoir résoudre dans des temps raisonnables des instances au-dessus de  $n = 50$  (avec  $m = n$ ). Étant donné que les instances provenant de la réalité ont plus d'un million de variables, il s'avère donc que la formulation  $(PCS_1)$  est définitivement une mauvaise stratégie pour résoudre  $(PCS)$ . Par contre, la dégénérescence reste négligeable lorsqu'on résout nos instances avec la formulation  $(PCS_2)$ .

Finalement, signalons que pour une instance donnée, la qualité de la borne inférieure se détériore avec l'augmentation de  $p$  et ce que l'on utilise  $(PCS_1)$  ou  $(PCS_2)$ .

### 3.3 Comparaison avec la littérature

La méthode proposée par Hansen, Labbé et Minoux est un algorithme dual qui utilise la formulation  $(PCS_2)$  avec un branchement dichotomique sur une variable du problème maître alors que pour notre part, il s'agit d'une méthode primale avec un branchement sur le problème auxiliaire. Étant donné de l'approche fort différente de ces deux méthodes, une comparaison autre que globale entre celles-ci resterait douteuse et imprécise. Pour cette raison, nous avons incorporé à notre algorithme la possibilité d'effectuer une énumération implicite par un branchement du type dichotomique (soit en posant  $y_k = 1$  et  $y_k = 0$ ) sur le  $(PM)$ . Il s'agit donc d'une approche intermédiaire qui va nous permettre de comparer l'efficacité entre l'approche duale et primale ainsi que d'évaluer l'impact du type de branchement sur la résolution.

Le tableau 3.2 présente les résultats de la comparaison avec Hansen, Labbé et Minoux. Les instances considérées sont données par  $n$  et  $p$  (avec  $m = n$ ,  $\delta = 2$

et  $\rho = 1$ ). La section  $PL_2$  du tableau 3.2 concerne les conditions de résolution de la relaxation linéaire à la racine de l'arbre de branchement. Cette section est commune aux 3 méthodes qui sont comparées ici. La section *ÉNUMÉRATION IMPLICITE* concerne les conditions de résolution sur l'ensemble de la résolution. Le terme *PRIMALE* réfère à l'approche primale et le terme *DUALE* à l'approche duale. De même, *Branch. sur (PA)<sub>j</sub>* indique que l'approche utilise un branchement sur le problème auxiliaire alors que *Branch. sur (PM)* signifie un branchement sur le problème maître. La colonne *temps de calcul* donne en secondes le temps consacré à la résolution du  $(PL_2)$  si elle se trouve dans la section  $PL_2$  et le temps total pour obtenir la solution optimale en nombres entiers si elle se trouve dans la section *ÉNUMÉRATION IMPLICITE*. La lettre  $\mu$  indique la moyenne du temps de calcul pour les instances considérées et  $\sigma$  correspond à l'écart type associé à  $\mu$ . L'écart

Tableau 3.2 Comparaison avec la littérature

n	p	PL <sub>2</sub>		ÉNUMÉRATION IMPLICITE							
		temps de calcul (s)		PRIMALE Branch. sur (PA) <sub>j</sub>			PRIMALE Branch. sur (PM)			DUALE Branch. sur (PM)	
				temps de calcul (s)			temps de calcul (s)			temps de calcul VAX8650 (s)	
				nœuds	$\mu$	$\sigma$	nœuds	$\mu$	$\sigma$	nœuds	$\mu$
10	2	0,01	0,009	3,4	0,05	0,02	4,6	0,07	0,03	1,3	1,0
	3	0,01	0,008	4,6	0,06	0,03	8,8	0,12	0,08	1,7	2,0
	5	0,01	0,006	8,0	0,10	0,04	18,6	0,24	0,14	3,2	2,0
20	3	0,03	0,008	4,8	0,11	0,09	5,0	0,13	0,14	1,6	6,0
	5	0,02	0,008	7,4	0,16	0,10	12,0	0,25	0,22	2,7	16,0
	10	0,02	0,006	24,4	0,43	0,19	87,6	1,64	1,00	6,3	43,0
30	5	0,04	0,011	10,0	0,34	0,16	15,6	0,49	0,52	5,6	95,0
	10	0,04	0,008	23,6	0,63	0,46	33,2	0,86	0,42	18,8	473,0
	15	0,03	0,006	50,6	1,27	0,69	124,0	3,38	3,28	14,7	413,0
40	5	0,07	0,014	6,0	0,35	0,36	5,0	0,25	0,20	3,8	583,0
	10	0,06	0,009	15,0	0,63	0,28	35,4	1,54	1,10	10,5	543,0
	15	0,06	0,012	30,4	1,11	0,69	75,2	2,94	2,58	25,3	1564,0
50	10	0,08	0,012	18,6	1,09	0,51	34,0	1,86	1,59	10,5	1110,0
	20	0,07	0,007	52,4	2,38	1,24	131,2	6,17	4,47	31,6	4806,0

type  $\sigma$  donne une appréciation de la stabilité des conditions de résolution d'une instance à l'autre. Le nombre total de nœuds visités (incluant la racine) dans l'arbre de branchement se trouve dans la colonne *nœuds*.

Le tableau 3.2 montre que les instances qui pouvait être considérées difficiles à résoudre par la méthode de Hansen, Labbé et Minoux deviennent faciles avec notre approche. Ceci s'explique par le fait que dans la méthode proposée par ces derniers auteurs, le problème auxiliaire (soit celui de génération de colonnes) consiste à résoudre un problème hyperbolique sous une contrainte. Or Hansen, Jaumard et Poggi de Aragão (1992) ont montré que ce problème est NP-dur alors que le problème auxiliaire de la méthode primale est polynomiale (proposition 2.2) que l'on prenne le branchement dichotomique sur le (PM) ou le branchement défini par (2.8) et (2.9) sur le problème auxiliaire.

Cette facilité de résolution du problème auxiliaire amène par contre un arbre de branchement plus important. Or la facilité du problème auxiliaire de la méthode primale compense si bien cet inconvénient que globalement la méthode primale reste plus efficace que la méthode duale. On remarque que le choix de l'énumération implicite joue aussi sur la grosseur de l'arbre de branchement. Il semble ici qu'un branchement sur le problème auxiliaire est souvent préférable à un branchement sur le problème maître. Ceci permet de conclure à une amélioration de la résolution en passant d'une méthode duale à une méthode primale et qu'on peut encore ajouter à l'amélioration en prenant une règle de branchement plus équilibrée sur les informations qui sont fixées. Nous allons maintenant approfondir l'importance du branchement.

### 3.4 Performance de la règle de branchement

Comme au tableau 3.2, les instances considérées dans le tableau 3.3 appartiennent à des classes telles que  $m = n$ ,  $\delta = 2$  et  $\rho = 1$ . Les instances sont décrites dans le tableau par les colonnes  $n$  et  $p$ . Comme précédemment, les sections  $PL_2$  et *ÉNUMÉRATION IMPLICITE* réfèrent aux conditions de résolution de la relaxation à la racine et de l'ensemble de la résolution. Seuls les résultats se rapportant à l'approche primale sont donnés dans le tableau 3.3. On retrouve des résultats relatifs aux branchements sur le problème auxiliaire et sur le problème maître sous les termes *Branchement sur les  $(PA)_j$*  et *Branchement sur le  $(PM)$* . La colonne *saut* donne les sauts de dualité observés en moyenne lors de la résolution des instances. On trouve dans la colonne *sol. ent.* le nombre d'instances dans chaque classe considérée ayant donné une solution optimale en nombres entiers dès la résolution de la relaxation linéaire à la racine de l'arbre de branchement. La colonne *cols* de la section  $PL_2$  donne le nombre de colonnes additionnelles devant être générées au cours de la résolution du  $(PL_2)$  alors que la colonne *cols* de la section *ÉNUMÉRATION IMPLICITE* fournit plutôt le nombre de colonnes générées en moyenne lors de la résolution du  $(PM)$  à un nœud donné de l'arbre de branchement autre que la racine. La profondeur maximale des arbres de branchement en nombre de nœuds est donnée dans la colonne *prof.* La présence de tirets dans le tableau 3.3 indique que les expériences n'ont pas été réalisées pour les valeurs de  $p$  correspondantes puisqu'elles ont été jugées trop coûteuses (on estime un temps de résolution supérieur à 50 000 secondes). Les autres colonnes du tableau 3.3 gardent la même signification qu'au tableau 3.2.

Il ressort du tableau 3.3 qu'une résolution avec notre branchement sur le problème auxiliaire est plus efficace qu'avec le branchement dichotomique sur une variable du

Tableau 3.3 Comparaison de la performance des règles de branchement

n	p	saut (%)	sol. ent.	PL <sub>2</sub>			ÉNUMÉRATION IMPLICITE									
				temps de calcul (s)			Branchement sur les (PA) <sub>j</sub>					Branchement sur le (PM)				
							temps de calcul (s)					temps de calcul (s)				
				cols	μ	σ	nœuds	prof	cols	μ	σ	nœuds	prof	cols	μ	σ
100	10	0,9	2	4,9	0,47	0,22	12,6	4,8	5,5	2,3	1,5	18,6	9,6	4,5	3,4	3,3
	20	2,9	0	2,9	0,29	0,12	39,2	10,6	5,0	5,5	2,1	60,8	29,9	3,2	8,5	4,1
	40	11,4	0	2,3	0,26	0,04	87,6	21,0	4,4	9,9	4,7	443,2	222,1	3,0	48,4	19,8
250	10	1,0	1	6,6	3,4	0,9	12,8	5,6	5,9	14	7	34,0	17,5	4,3	36	44
	25	0,3	1	3,4	2,1	0,7	10,2	5,1	5,6	11	9	32,2	16,6	4,1	30	48
	50	2,9	0	2,6	1,7	0,3	70,6	16,5	6,3	62	21	193,4	97,2	3,1	123	64
	75	7,4	0	2,6	1,7	0,3	146,8	34,9	4,6	101	42	1018,6	509,8	3,1	608	219
	100	14,0	0	2,6	1,6	0,3	343,2	60,0	4,3	218	61	3880,2	1940,6	3,2	2439	292
500	10	0,2	6	9,4	19,6	4,9	5,4	2,7	5,9	34	24	9,2	5,1	4,4	44	33
	50	0,5	0	2,4	8,0	2,0	22,0	8,2	6,9	86	62	120,2	60,6	3,6	427	798
	100	3,2	0	1,9	7,2	1,1	131,2	31,6	6,7	431	187	832,0	416,5	3,1	1801	711
	150	8,8	0	1,9	7,4	1,1	444,8	74,2	5,4	1192	386	10551,2	5204,6	3,2	18329	18131
	200	14,7	0	1,9	7,3	1,1	720,6	119,5	4,6	1625	385	-	-	-	-	-
1000	10	0,3	3	11,4	122	32	11,2	4,8	7,3	267	166	20,2	10,6	5,7	389	334
	100	0,2	1	3,7	65	27	40,4	10,9	7,1	614	582	74,6	37,8	4,4	1057	1288
	200	2,7	0	2,1	37	2	196,0	50,2	7,3	2649	1211	2826,8	1413,9	3,4	27020	8894
	300	8,2	0	2,1	39	3	843,4	131,5	5,7	9651	1989	-	-	-	-	-
1500	10	0,2	2	15,6	459	117	11,4	4,7	6,7	773	247	17,4*	9,2*	6,7*	1115*	772*
	150	0,3	0	4,5	156	57	104,8	32,3	6,8	3208	1694	607,8	270,4	4,2	15798	17229
	300	2,4	0	2,4	100	12	254,2	65,3	7,7	8166	1542	-	-	-	-	-
	450	7,9	0	2,4	98	12	1603,4	184,5	5,9	41008	11336	-	-	-	-	-

\* valeurs obtenues à partir de 9 instances au lieu de 10.



problème maître: les temps de calcul sont plus courts, les conditions de résolution plus stables (d'après  $\sigma$ ) et les arbres de branchement ont une profondeur moindre et contiennent moins de nœuds.

La stratégie adoptée pour le branchement dichotomique sur une variable  $y_k$  du (PM) consiste à explorer en premier le fils fixant  $y_k = 1$  puis ensuite le fils fixant  $y_k = 0$ . Cette stratégie a été arrêtée suite à l'observation que le branchement fixant  $y_k = 1$  amenait toujours une solution réalisable en nombres entiers. Ceci entraîne donc un invariant dans l'ordre que les nœuds sont visités dans l'arbre de branchement puisqu'à toute profondeur de l'arbre, il n'existe que deux nœuds. Ainsi, la comparaison des deux types de branchement du tableau 3.3 s'en trouve facilitée. Notons que le branchement imposant  $y_k = 0$  est d'une grande pauvreté comme le montre l'importance de la profondeur des arbres.

L'arbre développé avec le branchement sur le problème auxiliaire est fort différent. Comme discuté aux sections 2.5.4 et 2.5.6, les arbres de branchement sont effectivement profonds et peu larges. Pour des sauts de dualité faibles (moins de 1%), on retrouve la même proportion de nœuds et de profondeur que les arbres obtenus au paragraphe précédent, soit un nombre de nœuds près du double de la profondeur. Or contrairement aux arbres décrits plus haut, nous n'avons pas systématiquement pour l'un des côtés du branchement une solution réalisable en nombres entiers. En fait, les solutions résultant des branchements sont pratiquement toujours fractionnaires (ce qui implique qu'il n'y a pas d'invariant comme précédemment). La faible quantité de nœuds par rapport à la profondeur provient de la stratégie de branchement adopté à la section 2.5.6 qui nous amène à une première solution en nombres entiers qui se révèle souvent optimale, ce qui permet de sonder rapidement le reste de l'arbre. Malheureusement, l'augmentation du rapport entre le nombre de nœuds et la profondeur pour des sauts de dualité supérieurs à 1% montre qu'on ne peut

pas garantir l'optimalité de la première solution en nombres entiers. Par contre, puisque l'exploration de l'arbre reste minime comparativement au nombre total de nœuds dans l'arbre, on peut dire que notre stratégie de branchement nous amène néanmoins dans un voisinage très proche de la solution optimale. De plus, signalons que la sélection des usagers telle que proposée à la section 2.5.2 (soit l'application des branchements (2.8) et (2.9) sur deux usagers critiques) a toujours pu être réalisée en pratique bien qu'aucun résultat théorique puisse le garantir.

Pour des sauts de dualité équivalents, on remarque dans le tableau 3.3 que le nombre de nœuds (ainsi que la profondeur des arbres) augmente pour les deux types de branchement avec l'augmentation de  $n$ . Notons qu'il en est de même lorsque  $p$  augmente et que  $n$  reste fixe. En considérant le nombre moyen de colonnes générées à chaque nœud de l'arbre (soit la colonne  $\overline{cols}$ ), on s'aperçoit que la difficulté de résolution d'un nœud reste pratiquement identique quelque soit les valeurs de  $n$  et de  $p$ . En fait, il y a une légère diminution de la difficulté avec l'augmentation de  $p$ .

D'autre part, la croissance du temps de calcul  $\mu$  des deux types de branchement suit un comportement monotone non linéaire en fonction de  $p$ . Plus particulièrement, la croissance de  $\mu$  avec le branchement sur le (PM) est très rapide et on observe qu'un ordre de grandeur est ajoutée à chaque fois que  $p$  est doublé. Par contre, la croissance est beaucoup moindre avec le branchement sur le problème auxiliaire.

Si on ne considère que la résolution du (PL<sub>2</sub>), on remarque alors une décroissance de  $\mu$  lorsque  $p$  augmente ainsi qu'une plus grande stabilité dans les conditions de résolution (selon  $\sigma$ ). Ainsi, plus le saut de dualité augmente, plus sa résolution est facilitée. On a donc besoin de générer de moins en moins de colonnes avec l'augmentation du saut de dualité jusqu'à un cas limite où il n'y a plus de diminution du nombre de colonnes devant être générées. La limite semble se trouver autour d'un

saut de 3%. La raison de ce phénomène s'explique par le fait que l'on trouve à partir de cette limite toujours la même solution optimale du (PL<sub>2</sub>) quelque soit la valeur de  $p$ . Cette solution est fractionnaire et s'obtient sans manifestation notable de la dégénérescence. Par contre, sous les 3% de saut de dualité, il y a possibilité d'obtenir une solution optimale du (PL<sub>2</sub>) en nombres entiers. Cette possibilité s'accompagne d'une augmentation du nombre de colonnes à générer ainsi qu'une plus grande manifestation de la dégénérescence.

### 3.5 Effet du degré d'intersection entre $I$ et $J$

Jusqu'à présent, nous avons considéré que l'aspect de l'augmentation de la densité des ensembles  $I$  et  $J$  sur le carré  $[0, 1000]^2$ . Nous allons maintenant étudier sur ce même carré les conditions de résolution lorsque  $I$  et  $J$  ne sont plus confondus ( $\rho \neq 1$  quoique  $m = n$ ). L'intérêt de cette question provient du fait qu'il ne faut pas s'attendre en général à considérer le site occupé par un usager réel comme étant nécessairement un site potentiel pour l'ouverture d'une facilité. Ainsi, tous les degrés d'intersection possibles entre les ensembles  $I$  et  $J$  semblent admissibles par la réalité.

Nous reportons dans le tableau 3.4 des résultats seulement pour l'approche primale avec le branchement sur le problème auxiliaire. On reprend dans ce tableau les colonnes du tableau 3.3. Les instances considérées dans le tableau 3.4 sont décrites par les colonnes  $n$ ,  $p$  et  $\rho$  ( $m = n$  et  $\delta = 2$ ). La colonne *CPLEX* donne la proportion du temps total de résolution employée pour effectuer des pivots de l'algorithme du simplexe.

On observe dans le tableau 3.4 les deux cas extrêmes suivants: pour  $\rho = 0$ , on obtient

Tableau 3.4 Conditions de résolution selon le degré d'intersection  $\rho$  entre  $I$  et  $J$ 

n	p	ρ	saut (%)	sol. ent.	PL <sub>2</sub>			ÉNUMÉRATION IMPLICITE					
					cols	μ	σ	temps de calcul (s)			temps de calcul (s)		
100	20	0,00	0,0	10	16,2	0,47	0,11	1,0	1,0	-	0,47	0,11	68,0
		0,50	0,0	10	20,5	0,60	0,15	1,0	1,0	-	0,60	0,15	66,5
		0,75	0,6	8	20,0	0,57	0,15	5,8	2,7	4,7	1,09	1,08	63,2
		0,90	1,5	1	13,4	0,57	0,15	14,2	6,2	5,2	2,5	2,2	71,1
		0,96	1,8	0	9,0	0,58	0,17	24,4	8,6	4,7	4,5	2,8	68,6
		1,00	2,9	0	2,9	0,29	0,12	39,2	10,6	5,0	5,5	2,1	71,1
500	100	0,00	0,0	10	34,4	28,0	4,0	1,0	1,0	-	28	4	67,1
		0,50	0,0	10	39,4	31,9	4,8	1,0	1,0	-	32	5	67,2
		0,75	0,0	10	42,6	34,0	3,7	1,0	1,0	-	34	4	67,0
		0,90	0,1	8	36,4	29,3	2,1	4,0	2,5	5,6	36	17	58,1
		0,96	1,3	0	23,3	32,4	6,1	87,0	23,2	6,3	316	233	34,0
		1,00	3,2	0	1,9	7,2	1,1	131,2	31,6	6,7	431	187	41,9
1000	200	0,00	0,0	10	42,9	148	14	1,0	1,0	-	148	14	66,5
		0,50	0,0	10	55,0	185	31	1,0	1,0	-	185	31	66,1
		0,75	0,0	10	57,5	191	12	1,0	1,0	-	191	12	65,8
		0,90	0,0	10	50,7	167	23	1,0	1,0	-	167	23	65,6
		0,96	0,3	4	43,8	182	27	26,8	9,0	5,3	483	489	37,0
		1,00	2,7	0	2,1	37	2	196,0	50,2	7,3	2649	1211	35,5

une solution optimale en nombres entiers dès que la première relaxation linéaire est résolue mais toujours sous une manifestation importante de dégénérescence puisque seulement 3 ou 4 des colonnes générées permettent un déplacement non nul vers l'optimalité alors que pour  $\rho = 1$ , on obtient une solution optimale fractionnaire du (PL<sub>2</sub>) sans le moindre signe de dégénérescence. Entre ces deux extrêmes, il y a une zone de transition où l'on passe avec l'augmentation de  $\rho$  d'une solution optimale en nombres entiers du (PL<sub>2</sub>) à une solution optimale fractionnaire. De plus, si on fixe  $n$  et  $p$ , on observe que l'augmentation de  $\rho$  amène aussi une augmentation du saut de dualité et donc, de l'importance de l'énumération implicite.

Il semble que plus l'instance est dense sur le carré  $[0, 1000]^2$  (donc plus  $n$  est élevé), plus le degré  $\rho$  doit avoir une valeur importante pour rendre possible l'apparition d'une solution optimale fractionnaire du (PL<sub>2</sub>). Ainsi, plus l'instance est dense, plus

la zone de transition s'approche de  $\rho = 1$  et plus elle s'amincit. La position de la zone de transition dépend donc des valeurs de  $n$  et de  $p$ , mais aussi d'une relation qui est manifestement non linéaire.

Pour  $\rho < 1$ , la résolution de la relaxation ( $PL_2$ ) se présente plus difficile (en terme du nombre de colonnes générées) que la résolution du (PM) à tout autre nœud de l'arbre. En fait, quelque soit l'instance considérée, la résolution du (PM) à un nœud autre que la racine nécessite seulement la génération de 5 à 6 colonnes en moyenne. Comme discuté à la section 2.5.4, le branchement sur deux usagers critiques semble bien conserver d'un nœud à l'autre ce qui a été acquis précédemment lors de l'exploration de l'arbre. D'autre part, remarquons qu'une résolution du ( $PL_2$ ) donnant une solution optimale en nombres entiers passe en moyenne autour de 66% du temps à effectuer des pivots de l'algorithme du simplexe indépendamment de la grosseur de l'instance.

Le choix de Hansen, Labbé et Minoux de présenter en 1994 des résultats que pour des instances du types  $m = n$ ,  $\rho = 1$  semble être justifié par le fait qu'il s'agit des instances qui amènent le plus grand saut de dualité, qui sont donc les plus difficiles à résoudre. Ceci reste vrai si on suppose que le niveau de dégénérescence est le même pour toutes les instances. Or si l'on considère la dégénérescence comme un critère de la difficulté de résolution, il faudrait aussi s'intéresser aux instances pouvant fournir avec une forte présomption une solution optimale en nombres entiers dès la racine de l'arbre de branchement ( $\rho \leq 0,5$ ).

### 3.6 Effet de la dimension des instances

Nous allons maintenant poursuivre avec l'effet de la dimension des instances sur les conditions de résolution. Les instances sont identifiées dans le tableau 3.5 par  $n$ ,  $p$  et  $\delta$  (la dimension de l'instance) et seules les instances des classes  $m = n$  et  $\rho = 1$  sont considérées. La structure du tableau 3.5 reste identique à celle du tableau 3.4. Comme toutes les instances considérées ont fourni une solution optimale fractionnaire du (PL<sub>2</sub>), nous avons omis de mettre la colonne *sol. ent.* dans le tableau 3.5.

Tableau 3.5 Conditions de résolution selon la dimension des instances

$n$	$p$	$\delta$	saut (%)	PL <sub>2</sub>			ÉNUMÉRATION IMPLICITE					
				cols	temps de calcul (s)		temps de calcul (s)		CPLEX			
					$\mu$	$\sigma$	nœuds	prof	cols	$\mu$	$\sigma$	(%)
100	20	2	2,9	2,9	0,29	0,12	39,2	10,6	5,0	5,5	2,1	71,1
		3	3,6	1,3	0,21	0,02	22,0	7,4	4,7	3,4	2,4	69,1
		5	4,6	1,2	0,21	0,02	29,0	11,2	3,9	4,0	3,3	68,5
		10	7,6	1,1	0,20	0,01	46,0	16,9	4,0	6,6	3,7	71,7
		15	8,1	1,0	0,20	0,01	55,6	19,0	3,9	7,8	4,5	72,4
500	100	2	3,2	1,9	7,2	1,1	131,2	31,6	6,7	431	187	41,9
		3	3,5	1,4	6,7	0,6	109,0	32,8	5,1	304	183	41,8
		5	4,6	1,1	6,4	0,5	77,2	37,1	3,5	165	40	39,5
		10	6,5	1,0	6,3	0,1	148,8	67,7	3,6	343	104	44,5
		15	8,2	1,2	7,4	0,9	194,8	78,3	3,3	516	454	40,8
1000	200	2	2,7	2,1	37,1	2,4	196,0	50,2	7,3	2649	1211	35,5
		3	3,8	1,6	34,2	2,4	178,2	65,5	5,9	2187	1086	37,6
		5	5,0	1,4	34,1	2,4	194,0	84,4	4,0	2140	1002	45,2
		10	6,3	1,4	33,6	2,5	239,2	113,8	3,0	1915	647	36,8
		15	7,8	1,4	37,2	3,0	314,4	149,6	3,2	3503	3350	41,9

Observons que l'augmentation de  $\delta$  entraîne une augmentation de la valeur du saut de dualité. Comme discuté dans le cadre du tableau 3.3, la résolution du (PL<sub>2</sub>) à la racine de l'arbre se facilite (en termes de colonnes à générer) avec l'augmentation du saut de dualité. Le temps de calcul consacré à la résolution du (PL<sub>2</sub>) reste par contre relativement stable et il semble aucunement perturbé par la valeur de  $\delta$ .

On observe d'ailleurs un comportement identique pour la proportion du temps total moyen utilisé pour effectuer les pivots de l'algorithme du simplexe sur l'ensemble de la résolution (voir la colonne *CPLEX*). D'autre part, on retrouve une diminution semblable à celle du tableau 3.3 du nombre moyen de colonnes à générer pour résoudre le (PM) à chacun des nœuds de l'arbre lorsque la dimension  $\delta$  augmente. De plus, notons le phénomène suivant: pour un  $n$  et un  $p$  donnés, il existe un  $\delta$  pour lequel la résolution globale est plus aisée et qui ne correspond pas nécessairement à des arbres de branchement plus petits. Il semblerait que pour ce  $\delta$ , la dégénérescence serait minimisée. On obtiendrait probablement alors une plus grande différenciation des coûts associés à l'affectation des usagers.

## CHAPITRE 4

### Complément et différentes extensions

Notons que la formulation  $(PCS_2)$  représente une certaine idéalisation de la réalité puisqu'elle considère, par exemple, que l'équipement des facilités sont des antennes omnidirectionnelles dont on peut choisir la puissance d'émission ou que le lieu géographique n'offre aucun obstacle à la propagation des ondes. Or la réalité est loin d'être aussi parfaite. En effet, la puissance des équipements est généralement fixe et il n'y a que quelques modèles disponibles sur le marché, la variation du coût selon la puissance n'est habituellement pas une fonction linéaire et la géographie du terrain est d'ordinaire truffée d'obstacles. Nous allons donc considérer dans le présent chapitre différentes façons d'intégrer quelques aspects de la réalité sans modifier pour autant la formulation  $(PCS_2)$  ni la méthode de résolution présentée au chapitre 2.

#### 4.1 Extension aux coûts d'affectation non linéaires

Comme la détermination du coût d'une affectation revient au problème auxiliaire et non au problème maître, l'extension à des fonctions coûts non linéaires s'obtient par une généralisation de la contrainte (2.6) de la formulation  $(PA)_j$ . Il suffit de remplacer  $d_{ij}$  par  $f(d_{ij})$  où  $f(\cdot)$  est une fonction quelconque telle que  $f(d_{ij}) \geq 0$  et où  $f(d_{ij}) = 0$  si  $d_{ij} = 0$ .

Afin d'étudier l'impact de la fonction coût sur les performances de l'approche primale



avec le branchement sur le problème auxiliaire, nous reprenons des expérimentations sous trois types de fonction non linéaire pour des classes d'instances telles que  $m = n$ ,  $\delta = 2$  et  $\rho = 1$  où  $\delta$  représente la dimension de l'instance et  $\rho$  le degré d'intersection entre les ensembles  $I$  et  $J$ . Afin de permettre la meilleure comparaison possible avec ce qui a été présenté au chapitre 3, nous avons considéré les mêmes instances ainsi que les mêmes matrices  $K'$  de départ utilisées alors. Les nouvelles conditions de résolution sont comparées avec les sections  $PL_2$  et *Branchement sur les*  $(PA)_j$  du tableau 3.3.

#### 4.1.1 Fonctions du type logarithmique et exponentiel

La figure 4.1 donne les fonctions coûts continues non linéaires utilisées pour réaliser nos expériences. L'abscisse de la figure 4.1 représente la distance de l'utilisateur  $i$  au

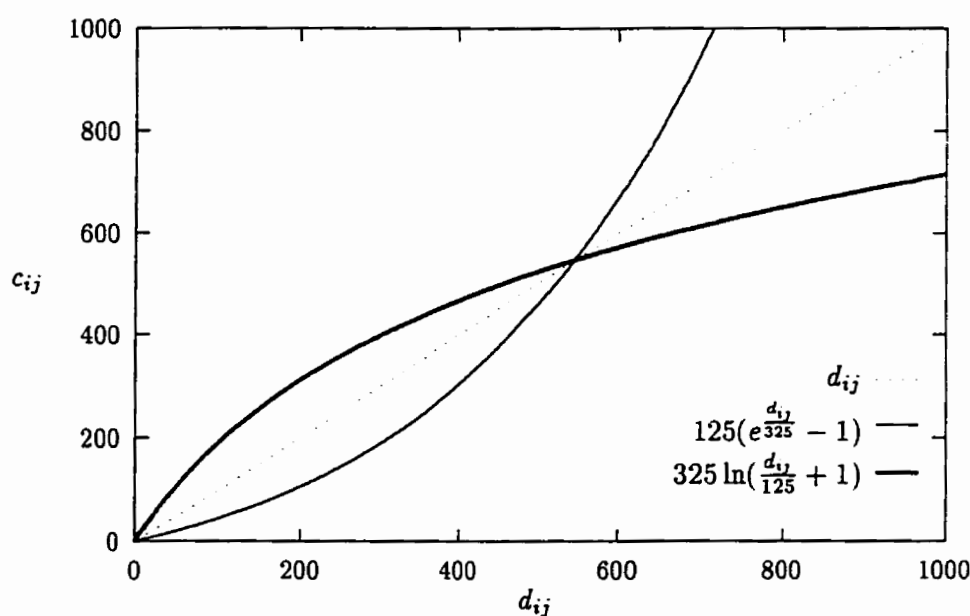


Figure 4.1 Fonctions logarithmique et exponentielle utilisées pour les expériences

site potentiel  $j$  ( $d_{ij}$ ) alors que l'ordonnée correspond au coût qu'il faut défrayer pour desservir une clientèle jusqu'à l'utilisateur  $i$  à partir d'une facilité ouverte en  $j$  ( $c_{ij}$ ). Les deux fonctions de notre étude ont été choisies de manière à être plus avantageuses sur un intervalle donné que la fonction linéaire  $f(d_{ij}) = d_{ij}$  considérée dans les expérimentations du chapitre 3 et plus dispendieuses ailleurs. La fonction logarithmique modélise une situation où l'augmentation de la portée du service nécessite une modification peu coûteuse des équipements de la facilité comparative-ment aux investissements antérieurs. La fonction exponentielle représente par contre une situation où l'augmentation de la portée du service signifie l'injection de grosses sommes d'argent parce que la technologie n'est pas disponible, par exemple.

Les tableaux 4.1 et 4.2 contiennent les résultats pour les deux fonctions non linéaires de la figure 4.1. Ces deux tableaux reprennent la structure du tableau 3.4. Les colonnes  $n$  et  $p$  décrivent les instances considérées lors des expérimentations (où  $m = n$ ,  $\delta = 2$  et  $\rho = 1$ ). La colonne *DEG* a été ajoutée afin de montrer la proportion des colonnes générées au cours de la résolution qui n'amènent aucun déplacement non nul vers l'optimalité. Cette colonne permet donc une appréciation du niveau de

Tableau 4.1 Conditions de résolution pour la fonction logarithmique

$n$	$p$	saut (%)	sol. ent.	PL <sub>2</sub>			ÉNUMÉRATION IMPLICITE						
				temps de calcul (s)			temps de calcul (s)						
				cols	$\mu$	$\sigma$	nœuds	prof	cols	$\mu$	$\sigma$	CPLEX (%)	DEG (%)
250	10	0,68	0	1,9	2,2	0,8	15,4	5,9	5,0	25	14	33,3	29,8
	50	9,1	0	1,3	1,8	0,3	143,2	38,3	4,6	160	33	30,7	41,7
	100	25,8	0	1,3	1,8	0,3	509,4	96,6	3,9	437	72	30,0	41,4
500	10	0,09	5	3,6	18,5	6,4	6,2	3,2	4,8	54	44	41,8	39,8
	100	9,6	0	1,1	8,5	0,7	325,0	72,8	5,2	1464	446	27,2	44,3
	200	26,4	0	1,1	8,4	0,7	1136,0	187,5	4,1	3771	770	23,0	40,8
1000	10	0,14	3	4,9	103	24	10,8	4,8	5,8	387	250	34,2	39,7
	100	2,6	0	1,1	41	3	137,0	49,7	4,9	2707	590	21,0	43,1
	200	9,4	0	1,1	40	3	657,6	148,8	5,0	11297	2852	23,0	45,7

Tableau 4.2 Conditions de résolutions pour la fonction exponentielle

n	p	saut (%)	sol. ent.	PL <sub>2</sub>			ÉNUMÉRATION IMPLICITE							
				cols	μ	σ	nœuds	prof	cols	μ	σ	CPLEX (%)	DEG (%)	
250	10	0,17	4	99,5	63,8	34,3	4,6	2,6	41,4	131	98	53,5	73,2	
	50	0,90	1	14,1	9,3	4,3	41,6	12,4	9,7	112	114	39,5	64,1	
	100	2,7	0	8,5	5,7	1,8	123,0	28,8	5,8	168	82	30,8	55,4	
500	10	0,16	4	204,8	620	415	6,0	3,1	127,4	3534	3726	77,2	65,9	
	100	0,58	0	15,1	46	24	35,8	12,6	16,0	902	1377	63,1	67,6	
	200	1,2	0	15,5	39	48	324,6	55,3	7,0	1360	1328	28,5	61,5	

dégénérescence rencontré lors de la résolution.

La comparaison des résultats du tableau 4.1 avec ceux du tableau 3.3 montre que les sauts de dualité sont plus importants pour la fonction coût logarithmique que pour une fonction linéaire sauf pour de petites valeurs de  $p$  ( $p = 10$ ). On remarque d'ailleurs qu'avec la fonction logarithmique les arbres de branchement sont plus profonds et que le temps total de résolution est en moyenne deux fois plus long. Par contre, la proportion du nombre de nœuds par rapport à la profondeur des arbres reste pratiquement inchangé avec ce que l'on observe au tableau 3.3 pour une valeur donnée de  $p$ . De plus, ce que nous avons dit suite au tableau 3.3 sur la sélection des usagers, sur les conséquences d'une augmentation des valeurs de  $n$  et de  $p$  sur le branchement au problème auxiliaire ainsi que sur les conditions de résolution du (PL<sub>2</sub>) demeure valide pour la fonction coût logarithmique.

Le tableau 4.2 montre pour la fonction coût exponentielle des sauts de dualité nettement plus faibles que ceux donnés aux tableaux 3.3 et 4.1. D'ailleurs, on observe que les arbres de branchement sont plus petits avec la fonction exponentielle. En fait, leur profondeur peut se réduire jusqu'à la moitié de celle atteinte lors d'une énumération implicite avec la fonction coût linéaire. De plus, le nombre de nœuds

par rapport à la profondeur des arbres reste dans les mêmes proportions pour un  $p$  donné que celles observées précédemment avec les fonctions coûts linéaire et logarithmique. On pourrait donc s'attendre à résoudre plus rapidement une instance dont le coût des affectations suit une fonction exponentielle plutôt qu'une fonction linéaire ou logarithmique. Or ce n'est pas le cas puisque la dégénérescence atteint un niveau suffisamment élevé pour détériorer la qualité de la résolution.

Ainsi, on observe que plus  $p$  est petit, plus grand est le nombre de colonnes devant être générées pour résoudre le (PM) à tous les nœuds de l'arbre de branchement (incluant le  $(PL_2)$  de la racine). D'ailleurs, l'augmentation du nombre de colonnes pour résoudre  $(PL_2)$  devient très important pour  $p = 10$  comparativement à ce qui est donné aux tableaux 3.3 et 4.1. On observe aussi une augmentation du nombre de colonnes pour les autres nœuds de l'arbre. Cette augmentation reste toutefois moins importante. Par contre, contrairement aux fonctions coûts linéaire et logarithmique, on remarque que le nombre de colonnes générées à chaque nœud de l'arbre (en excluant la racine) varie avec la valeur de  $p$  (en fait, elle augmente lorsque  $p$  diminue). Ceci entraîne naturellement une augmentation des temps de résolution. De plus, la croissance du temps total de résolution n'est plus monotone avec l'augmentation de  $p$ .

L'origine de cette détérioration des conditions de résolution provient bien d'une manifestation trop importante de la dégénérescence lorsque  $p$  est petit. D'ailleurs, les colonnes *CPLEX* et *DEG* des tableaux 4.1 et 4.2 montrent que plus la valeur de  $p$  est petite avec la fonction coût exponentielle, plus la quantité de temps utilisée pour effectuer des pivots est importante et plus la proportion de colonnes générées ne permettant pas un déplacement non nul vers l'optimalité est grande. Notons que la génération des colonnes ayant un déplacement non nul lors de nos expériences n'a pas nécessité l'introduction de la règle d'anticyclage relative à la génération de

colonnes (voir la section 2.4).

#### 4.1.2 Fonctions du type par paliers

Nous allons considérer maintenant les fonctions coûts variant par paliers. Ces fonctions ont la capacité de modéliser plus judicieusement la réalité puisque la disponibilité des équipements pouvant constituer une facilité suit souvent une standardisation imposée par l'industrie qui ne produit qu'un ensemble discret d'équipements. Avec les fonctions continues, les listes  $L_j$  contenaient autant de sous-listes complètes maximales que d'utilisateurs lorsque chaque utilisateur  $i$  était à une distance strictement différente du site potentiel  $j$ . Avec les fonctions par paliers, nous aurons plutôt autant de sous-listes complètes maximales que de paliers pour rejoindre le dernier utilisateur de  $L_j$ . Ceci signifie que l'utilisation la plus rentable d'un équipement est de l'utiliser à sa pleine capacité.

Les fonctions coûts par paliers utilisées dans les expérimentations sont données à la figure 4.2 où l'abscisse correspond à la distance séparant l'utilisateur  $i$  du site  $j$  et l'ordonnée au coût pour desservir  $i$  à partir de  $j$ . Afin d'illustrer l'effet du nombre de paliers sur la résolution, nous avons considéré des fonctions avec 3 et 5 paliers. Ces deux fonctions approchent une fonction exponentielle. Nous avons observé que les fonctions par paliers approchant une fonction linéaire donnait toujours une solution optimale du (PL<sub>2</sub>) en nombres entiers, ce qui n'est pas le cas pour une fonction exponentielle.

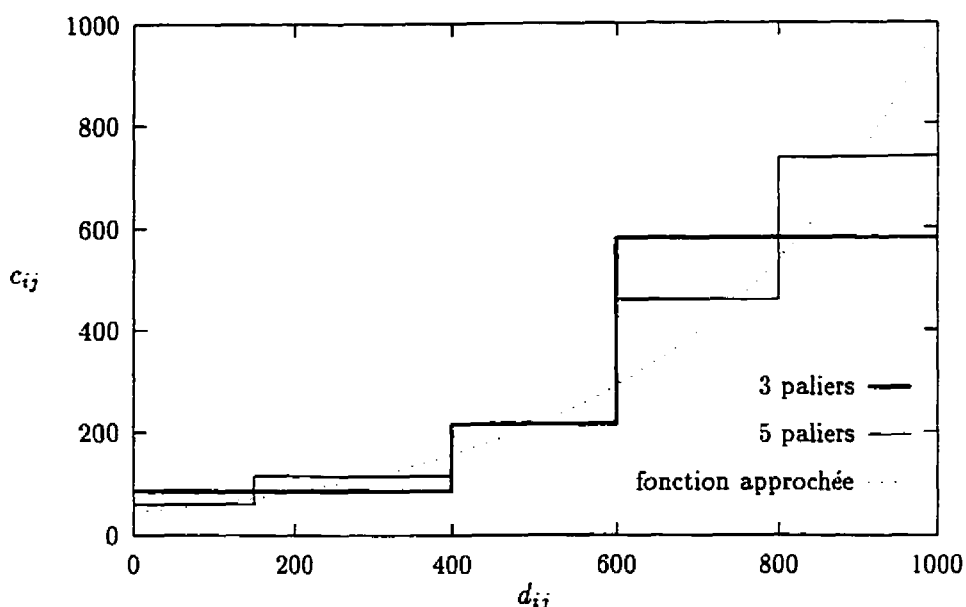


Figure 4.2 Fonctions par paliers utilisées pour les expériences

Les résultats des expérimentations sur la fonction approchée de la figure 4.2 sont donnés aux tableaux 4.3 et 4.4 pour des classes d'instances telles que  $m = n$ ,  $\delta = 2$  et  $\rho = 1$ . Ces deux tableaux ont exactement la même structure que les tableaux 4.1 et 4.2.

En comparant les tableaux 4.3 et 4.4, on constate que le saut de dualité, la difficulté du (PL<sub>2</sub>) en terme de colonnes à générer ainsi que l'importance de l'énumération implicite dépendent du nombre de paliers. En fait, plus il y a de paliers, plus les instances sont difficiles à résoudre. Une résolution avec la fonction de 5 paliers génère plus de colonnes et plus de nœuds qu'avec la fonction de 3 paliers. Par contre, la fonction de 3 paliers passe plus de temps à effectuer des pivots de l'algorithme du simplexe que la fonction ayant 5 paliers, ce qui indique que la fonction de 3 paliers amène une plus grande dégénérescence. Notons cependant que le nombre de colonnes générées ayant un déplacement nul vers l'optimalité est indépendant des valeurs de

Tableau 4.3 Conditions de résolution pour la fonction approchée par 3 paliers

n	p	saut (%)	sol. ent.	PL <sub>2</sub>			ÉNUMÉRATION IMPLICITE						
				temps de calcul (s)			temps de calcul (s)					CPLEX (%)	DEG (%)
				cols	$\mu$	$\sigma$	nœuds	prof	cols	$\mu$	$\sigma$		
250	2	0,0	10	15,9	4,2	1,3	1,0	1,0	-	4,2	1,3	60,4	81,4
	3	0,0	8	16,4	4,0	1,4	1,4	1,2	11,5	4,6	2,0	56,9	82,4
	5	0,0	8	33,4	7,9	1,8	1,4	1,2	5,5	8,3	2,3	59,1	86,6
	10	1,6	7	31,0	7,7	1,7	1,6	1,3	6,7	8,3	2,3	57,4	83,2
	50	1,6	7	31,0	7,3	1,7	1,6	1,3	6,7	7,9	2,2	57,4	83,2
500	2	0,0	10	23,2	24,6	13,8	1,0	1,0	-	24,6	13,8	62,8	85,6
	3	1,6	8	24,0	24,9	8,7	1,6	1,3	7,2	27,0	10,0	59,2	81,8
	5	0,0	10	58,5	62,6	20,6	1,0	1,0	-	62,6	20,6	65,5	86,7
	10	0,0	9	59,0	64,8	22,0	1,2	1,1	15,5	66,5	24,8	64,9	86,0
	100	0,0	9	59,0	63,3	21,7	1,2	1,1	15,5	64,8	23,8	64,7	86,0
1000	2	0,0	10	40,1	166	99	1,0	1,0	-	166	99	63,2	84,2
	3	0,0	10	40,7	175	91	1,0	1,0	-	175	91	64,8	86,1
	5	0,0	8	85,5	430	263	2,2	1,6	23,2	466	252	66,4	79,7
	10	0,0	8	79,4	389	260	2,2	1,6	66,8	623	648	66,9	78,3
	200	0,0	8	79,4	409	286	2,2	1,6	62,0	613	562	66,6	78,3

Tableau 4.4 Conditions de résolution pour la fonction approchée par 5 paliers

n	p	saut (%)	sol. ent.	PL <sub>2</sub>			ÉNUMÉRATION IMPLICITE						
				temps de calcul (s)			temps de calcul (s)					CPLEX (%)	DEG (%)
				cols	$\mu$	$\sigma$	nœuds	prof	cols	$\mu$	$\sigma$		
250	2	4,7	0	32,6	8,6	2,7	5,0	3,0	4,6	12,1	2,8	51,0	79,1
	3	5,6	5	38,3	9,8	2,9	5,6	2,9	8,5	15,0	7,4	50,3	80,1
	5	1,6	5	40,5	10,2	3,5	3,6	2,2	8,7	13,4	4,9	54,2	82,8
	10	1,6	5	40,5	10,5	3,6	3,8	2,2	8,9	13,8	5,5	53,8	82,7
	50	1,6	5	40,5	10,2	3,6	3,8	2,2	8,9	13,3	5,3	54,4	82,7
500	2	3,0	1	46,2	49,6	15,9	4,0	2,5	8,4	65	21	53,8	82,8
	3	1,9	5	52,2	55,8	14,6	6,2	2,9	12,7	84	37	50,1	78,7
	5	2,5	4	54,1	57,7	18,0	10,2	4,0	12,6	105	83	45,5	81,3
	10	2,5	4	54,1	58,8	18,8	10,8	3,8	13,4	111	94	44,8	81,8
	100	2,5	4	54,1	58,9	18,5	10,4	4,0	11,8	106	84	45,4	81,5
1000	2	0,9	4	83,3	381	116	2,2	1,6	13,4	413	118	64,5	82,6
	3	2,6	3	90,8	425	114	19,8	5,9	20,9	976	559	40,8	76,4
	5	2,6	3	99,9	462	133	25,4	6,9	23,8	1422	1037	41,1	74,9
	10	2,6	3	97,5	457	146	26,4	6,5	22,7	1291	948	39,6	75,4
	200	2,6	3	97,5	470	142	29,0	7,1	22,0	1405	1053	38,5	75,7

$n$  et de  $p$ , mais semble décroître avec l'augmentation du nombre de paliers. Quoiqu'il en soit, les valeurs dans les colonnes *CPLEX* et *DEG* des tableaux 4.3 et 4.4 montrent

que les instances dont le coût des affectations suit une fonction par paliers sont fortement dégénérées.

Remarquons que la difficulté de résolution du  $(PL_2)$  sous une fonction coût par paliers augmente avec la valeur de  $p$  alors qu'on observait le contraire pour les fonctions continues. Toutefois, comme discuté au tableau 3.3, on retrouve ici aussi une valeur de  $p$  pour laquelle il n'y a plus d'augmentation de la difficulté du  $(PL_2)$ . Ce  $p$  critique se trouve autour des valeurs 5 et 10. Notons de plus que les conditions de résolution sur l'ensemble de la résolution deviennent indépendantes de  $p$  pour des valeurs supérieures au  $p$  critique. Les conditions de résolution sont alors très stables.

D'autre part, comme on peut le constater dans le tableau 4.3, il faut s'attendre à rencontrer avec les fonctions coûts par paliers des solutions optimales fractionnaires du  $(PL_2)$  dont la valeur correspond à la solution optimale en nombres entiers. L'énumération ne sert alors qu'à trouver une configuration d'affectations en nombres entiers.

## 4.2 Intégration de contraintes à même les $L_j$ et les $E_j$

Par l'utilisation des vecteurs  $E_j$ , nous avons vu à la section 2.6 comment on pouvait éviter d'explicitier les contraintes de branchement dans la formulation du problème auxiliaire. Nous allons maintenant énumérer différentes façons d'incorporer d'autres types de contraintes à même les vecteurs  $E_j$  en initialisant certain  $e_k^j = 1$  dès le début de la résolution, ou à même les listes  $L_j$  en les tronquant ou en modifiant la valeur des  $d_{ij}$ . L'énumération qui suit s'applique à la version recouvrement du



problème du *p-Centre-Somme*.

1. S'il existe une portée maximale impossible à dépasser telle qu'on ne peut pas physiquement recouvrir tous les usagers de  $I$  à partir d'une seule facilité, on peut tenir compte de cette restriction en retirant de  $L_j$  tous les usagers  $i$  dont la distance au site  $j$  est plus grande que la portée maximale des équipements disponibles.
2. Si le nombre d'usagers pouvant être desservis par une facilité ouverte au site  $j$  est limité à  $w_j$ , il suffit de conserver les  $w_j$  premiers usagers de  $L_j$  pour satisfaire une telle contrainte. Selon la réalité à modéliser, on peut avoir besoin de tronquer d'avantage  $L_j$  surtout si on doit garder l'intégrité des sous-listes complètes maximales de la liste  $L_j$  originale.
3. On peut aussi simuler des services directionnels (afin de tenir compte, par exemple, des obstacles naturels, des corridors de services ou de l'orientation des antennes en téléphonie cellulaire) en retirant de  $L_j$  tous les usagers se trouvant hors de la zone de service. Notons que dans le cas des antennes directionnelles, on pourrait être intéressé par plusieurs directions possibles. En se donnant une liste  $L_j^d$  pour chaque orientation  $d$  possible de l'antenne au site  $j$ , on définirait ainsi un ensemble  $K_j^d$  d'affectations d'usagers pour chaque orientation  $d$ . Comme on aurait  $K_j = \bigcup_d K_j^d$ , l'introduction de la contrainte de convexité  $\sum_{k \in K_j} y_k \leq 1$  dans la formulation (PCS<sub>2</sub>) serait nécessaire afin d'assurer qu'une seule affectation d'usagers puisse être retenue au site  $j$  à l'optimalité. L'orientation de l'antenne en  $j$  serait alors déterminée par la variable  $y_k = 1$  où  $k$  appartient à l'un des  $K_j^d$ .
4. Par une initialisation judicieuse des valeurs  $d_{ij}$ , on peut considérer des fonctions de service non monotones où la portée du service varie selon son orientation.

5. L'indépendance des  $(PA)_j$  donne la possibilité d'associer à chaque site  $j$  une fonction coût d'affectation différente.
6. L'agrégation du coût total d'une affectation au coefficient  $d_{ij}$  du dernier usager  $i$  affecté au site  $j$  permet de modéliser toutes réalités telles que la présence d'un usager dans l'affectation implique nécessairement un coût (contrairement au modèle original où seul l'usager le plus éloigné amenait un coût), mais où l'affectation des usagers est cumulative à la manière du recouvrement (c.-à-d. qu'il existe un ordre dans lequel l'affectation des usagers doit être faite).

Ces 6 façons d'intégrer certaines contraintes à la version recouvrement du problème du *p-Centre-Somme* s'étendent trivialement à la version de mise en boîte (*packing*). Il en est toutefois autrement pour la version partition du problème du *p-Centre-Somme*, particulièrement en ce qui concerne les points 2 et 6 de l'énumération puisque l'aspect cumulatif de l'affectation des usagers n'existe plus alors.

### 4.3 Généralisation du modèle

Les extensions considérées jusqu'ici étaient essentiellement orientées vers la résolution des aspects de localisation et d'affectation du problème du *p-Centre-Somme*. Outre l'extension immédiate à la classification, la méthode de résolution présentée au chapitre 2 peut facilement s'appliquer à d'autres types de problèmes dont la formulation est proche de celle du  $(PCS_2)$ .

### 4.3.1 Nouvel algorithme pour le problème de Recouvrement

Si on pose  $p = m$ , la contrainte (1.12) du  $(PCS_2)$  sera trivialement satisfaite et le problème du  $p$ -Centre-Somme se réduira alors à trouver un recouvrement qui minimise la fonction objectif (1.10). Le  $(PCS_2)$  est une généralisation du problème classique du *Recouvrement*.

La méthode de résolution du chapitre 2 peut donc être utilisée pour résoudre le problème du *Recouvrement* au lieu des algorithmes de Balas (1980), de Balas et Ho (1980), de Fisher et Kedia (1990), et de Harche et Thompson (1994). La génération de colonnes suppose toutefois qu'il existe un oracle qui permet de construire chacune des colonnes de la matrice des contraintes (ce qui n'est pas le cas des méthodes citées ci-haut). Or on peut souvent élaborer un tel oracle pour la plupart des problèmes provenant de la réalité puisque l'existence d'une colonne dépend des relations entre les objets d'une réalité donnée et que ces relations sont souvent assez bien connues. Tel que discuté à la section 2.1, on pourrait ainsi s'attaquer à des problèmes de quelques millions de variables.

### 4.3.2 Application au domaine de localisation continue

Notre méthode s'applique également au domaine de localisation continu (soit le cas *discret/continu* du problème du  $p$ -Centre-Somme). À partir des indications données par Chen et Handler (1987) pour déduire un nombre fini de centres optimaux potentiels pour le problème du  $p$ -Centre dans le plan, on peut aisément construire un ensemble  $J$  discret utilisable par la méthode de résolution. On ramène ainsi le cas *discret/continu* au cas *discret/discret* que l'on sait résoudre.

### 4.3.3 Interprétation étendue du modèle

Les fonctions coûts par paliers ajoutent sans apporter aucune modification au ( $\text{PCS}_2$ ) une nouvelle dimension au problème du *p-Centre-Somme* tel que défini au premier chapitre. Ainsi, en plus de décider de l'affectation des usagers et de la localisation des facilités, on peut aussi déterminer l'équipement constituant les facilités qui seront ouvertes. Le choix de l'équipement à un site donné est fixé par le palier occupé par l'utilisateur critique de l'affectation au site en question.

Par l'introduction de contraintes de convexité dans la formulation ( $\text{PCS}_2$ ) et l'utilisation d'un ensemble de listes  $L_j^d$  pour chaque site  $j$ , on pourrait aller encore plus loin en spécifiant l'orientation qu'il faut donner aux équipements des facilités. De manière plus générale, les listes  $L_j^d$  ont la capacité de représenter plusieurs types de clientèles (en termes de besoin et de pouvoir d'achats). C'est donc dire qu'on pourrait par cette modification du ( $\text{PCS}_2$ ) choisir d'affecter à chaque facilité les usagers d'une clientèle donnée sans fixer d'avance cette clientèle.

D'autre part, lors de la présentation de la définition du problème du *p-Centre-Somme* telle que donnée par Hansen, Labbé et Minoux (voir section 1.1), nous avons fourni une interprétation du problème selon le point de vue des usagers, soit selon leur satisfaction naturelle d'être affectés à une facilité ouverte sur un site plutôt que sur un autre. On peut maintenant réinterpréter le problème du *p-Centre-Somme* du point de vue du fournisseur de services, lequel s'intéresse au coût qu'il faut déboursier pour mettre en place un certain service d'excellence. En fait, le fournisseur désire ici mettre en place les équipements nécessaires pour répondre aux exigences du client le plus difficile qu'il veut desservir. Comme le restant de la clientèle sera desservi de manière adéquate (les exigences étant moindres), le modèle (avec les modifications

discutées précédemment) permet donc au fournisseur de services de choisir le lieu d'ouverture de ses facilités en considérant leur coût d'implantation ainsi que les coûts fixes d'exploitation.

## CONCLUSION

Pour résoudre le problème du *p-Centre-Somme*, nous avons utilisé une méthode primale de résolution par génération de colonnes. Précédemment, un algorithme dual de génération de colonnes avait été suggéré par Hansen, Labbé et Minoux en 1994. Suite aux expérimentations numériques, l'approche primale s'est révélée beaucoup plus efficace pour résoudre le problème du *p-Centre-Somme* que l'approche duale. D'ailleurs, Hansen, Jaumard et Poggi de Aragão (1992) avaient déjà observé la supériorité de l'approche primale sur l'approche duale pour le problème de *Satisfaisabilité Probabiliste Maximale*. On doit l'efficacité de notre algorithme à certaines propriétés de l'arbre d'énumération sous les branchements (2.8) et (2.9) ainsi qu'à la complexité du problème auxiliaire qui reste polynomiale même sous ce branchement.

De plus, notre modèle (soient les formulations (PM) et  $(PA)_j$  incluant les listes  $L_j$  et les vecteurs  $E_j$ ) est relativement simple et flexible, il ne requiert pas l'utilisation d'une norme spécifique ni d'un type particulier de fonction pour la détermination du coût des affectations. La flexibilité d'utilisation des listes  $L_j$  et des vecteurs  $E_j$  permet aussi d'introduire dans notre modèle certaines contraintes de capacité et de passage sans en altérer la simplicité.

Notre algorithme de résolution est par contre sujet à deux comportements qui ont tendance à s'exclure mutuellement: soit que les arbres de branchement descendent à de grandes profondeurs, soit que la dégénérescence atteint un niveau suffisamment élevé pour détériorer les conditions de résolution du (PM). Afin d'éviter des arbres trop profonds, on pourrait rechercher une meilleure borne supérieure que celle

obtenue présentement. Ceci pourrait s'obtenir par une heuristique ou par une modification de la stratégie de branchement (profondeur d'abord en branchant en premier sur le meilleur des fils, par exemple). Une amélioration du critère de sélection des usagers pourrait aussi apporter une diminution de la profondeur des arbres. En ce qui concerne la dégénérescence, une première approche consisterait à introduire à chaque itération de l'Algorithme 2 toutes les colonnes de coût réduit négatif obtenues par la résolution des  $(PA)_j$ . On pourrait aussi reconsidérer la matrice initiale servant à amorcer l'Algorithme 2.

Quoiqu'il en soit, la méthode de résolution tel que présentée dans ce mémoire permet de résoudre des instances de plus de 2 millions de variables ( $m = n = 1500$  et  $p \leq 150$ ) en des temps raisonnables. De plus, notre méthode s'applique aisément au domaine de localisation continu.

Une extension intéressante à notre travail serait de considérer la modification d'un réseau de services existant par l'ajout (ou le retrait) de facilités afin de satisfaire la nouvelle demande. On pourrait aussi considérer comme variante le remplacement de l'équipement des facilités. Enfin, nous voudrions attirer l'attention sur le problème du *p-Centre-Somme* sous des contraintes de capacité (au sens d'une limitation du nombre d'usagers pouvant être affecté à chaque facilité) pour la version partitionnement. Étant donné l'absence de l'aspect cumulatif de l'affectation des usagers, est-il possible d'obtenir pour ce problème un algorithme de résolution tel que la complexité du problème auxiliaire reste polynomiale sous le branchement?

## BIBLIOGRAPHIE

BALAS, E. (1980). Cutting planes from conditional bounds: a new approach to set covering. *Mathematical Programming*, 12, 19-36.

BALAS, E., HO, A. (1980). Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study. *Mathematical Programming*, 12, 37-60.

BARNHART, C., JOHNSON, E.L., NEMHAUSER, G.L., SAVELSBERG, M.W., VANCE, P.H. (1994). Branch-and-prince: column generation for solving huge integer programs. *Mathematical Programming: State of the Art 1994*, J.R. Birge and K.G. Murty (editors), University of Michigan, 186-207.

BEALE, E.M.L., TOMLIN, J.A. (1970). Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. *Proceedings of the Fifth International Conference on Operational Research*, J. Lawrence (editor), Tavistock Publications, 447-545.

CHEN, R., HANDLER, G.Y. (1987). Relaxation method for the solution of the minimax location-allocation problem in euclidean space. *Naval Research Logistics*, 34, 775-788.

CHRISTOFIDES, N. (1975). *Graph Theory, an Algorithmic Approach*, Academic Press, London.

CHVÁTAL, V. (1983). *Linear Programming*, Freeman, New York.



DESROSIERS, J., SOUMIS, F., DESROCHERS, M. (1984). Routing with time windows by column generation. *Networks*, 14, 545-565.

FISHER, M.L., KEDIA, P. (1990). Optimal solution of set covering/partitioning problems using dual heuristics. *Management Science*, 36, 674-688.

HAMMER, P.L., HANSEN, P., SIMEONE, B. (1982). Vertices belonging to all or no maximum stable sets of a graph. *SIAM Journal Algebraic and Discrete Methods*, 4, 511-522.

HANSEN, P., JAUMARD, B., POGGI DE ARAGÃO, M. (1991). Un algorithme primal de programmation linéaire généralisée pour les programmes mixtes. *Comptes-rendus de l'Académie des Sciences de Paris*, 313, 557-560.

HANSEN, P., JAUMARD, B., POGGI DE ARAGÃO, M. (1992). Mixed-integer column generation algorithms and the probabilistic maximum satisfiability problem. *Integer Programming and Combinatorial Optimization*, E.Balas, G.Cornuéjols, R.Kannan (editors), Graduate School of Industrial Administration, Carnegie Mellon University, 165-180.

HANSEN, P., JAUMARD, B., SANLAVILLE, E. (1994). Weight constrained minimum sum-of-stars clustering. *Les Cahiers du GERAD*, G-93-38 (à paraître dans *Journal of Classification*).

HANSEN, P., MINOUX, M., LABBÉ, M. (1987). Extension de la programmation linéaire généralisée au cas des programmes mixtes. *Comptes-rendus de l'Académie des Sciences de Paris*, 305, 569-572.

HANSEN, P., LABBÉ, M., MINOUX, M. (1994). The  $p$ -Center-Sum location problem. *Cahiers du Centre d'Études de Recherche Opérationnelle*, 36, 203-219.

HARCHE, F., THOMPSON, G.L., (1994). The column subtraction algorithm: an exact method for solving weighted set covering, packing and partitioning problems. *Computers Operations Research*, 21, 689-705.

LASDON, S.L. (1970). *Optimization Theory for Large Systems*, MacMillan, New York.

NEMHAUSER, G.L., TROTTER L.E. (1975). Vertex packings: structural properties and algorithms. *Mathematical Programming*, 8, 232-248.

NEMHAUSER, G.L., WOLSEY L.A. (1988). *Integer and Combinatorial Optimization*, Wiley, New York.

PARKER, R.G., RARDIN, R.L. (1988). *Discrete Optimization*, Academic Press, San Diego.

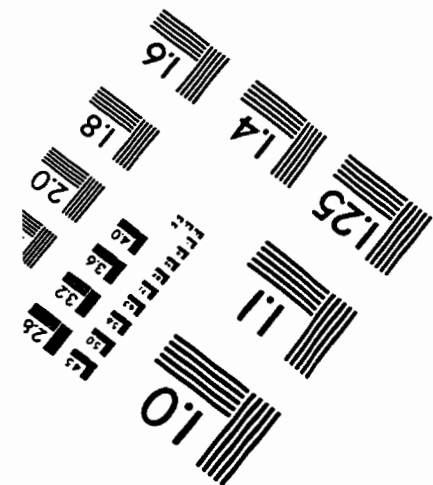
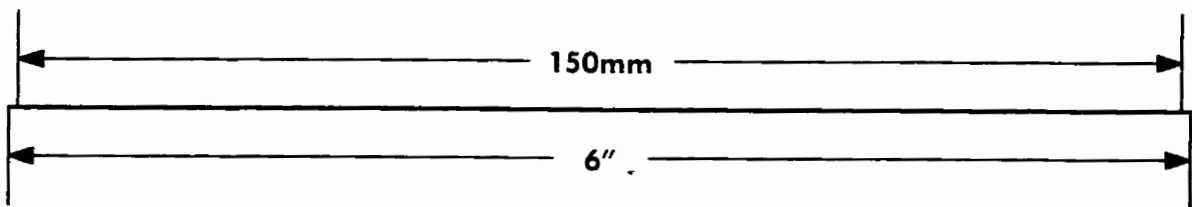
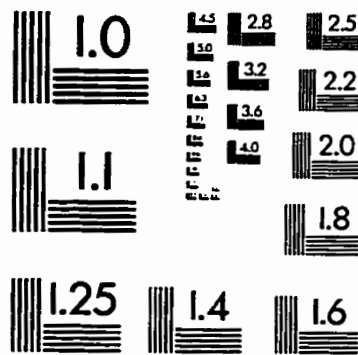
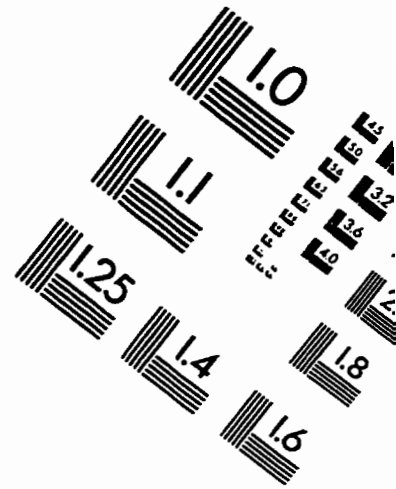
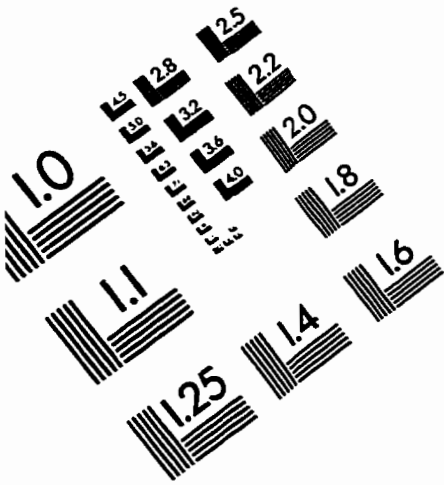
RYAN, D.M., FOSTER, J.C. (1981). An integer programming approach to scheduling. *Computer Scheduling of Public Transport*, A. Wren (editor), North-Holland, Amsterdam, 269-280.

TAHA, H.A. (1975). *Integer Programming: Theory, Applications, and Computations*, Academic Press, Orlando.

TRUSHIN, S.N. (1980). Finding the largest internally stable sets of a graph by successive analysis of alternatives. *Cybernetics*, 5, 669-679.

VANDERBECK, F., WOLSEY, L.A. (1996). An exact algorithm for IP column generation. *Operations Research Letters*, 19, 151-159.

# IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc.  
1653 East Main Street  
Rochester, NY 14609 USA  
Phone: 716/482-0300  
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved

